



HAL
open science

Étude et résolution exacte de problèmes de transport à la demande avec qualité de service

Thierry Garaix

► **To cite this version:**

Thierry Garaix. Étude et résolution exacte de problèmes de transport à la demande avec qualité de service. Modélisation et simulation. Université d'Avignon, 2007. Français. NNT : . tel-00534894

HAL Id: tel-00534894

<https://theses.hal.science/tel-00534894v1>

Submitted on 10 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ACADÉMIE D'AIX-MARSEILLE
UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE

THÈSE de DOCTORAT

Présentée pour obtenir le grade de Docteur en Sciences
de l'Université d'Avignon et des Pays de Vaucluse

Spécialité : Informatique, recherche opérationnelle et géomatique

*Étude et résolution exacte de problèmes de transport
à la demande avec qualité de service*

par
Garaix Thierry

Soutenue publiquement le 13 décembre 2007 devant un jury composé de :

M. Claramunt Christophe	Professeur, IRENav, Brest	Rapporteur
M. Cordeau Jean-François	Professeur, HEC Montréal	Rapporteur
M. Prins Christian	Professeur, Univ. Technologies Troyes	Rapporteur
M. Artigues Christian	Chargé de recherches (HDR), LAAS-CNRS, Univ. Toulouse	Co-directeur
M. Charre Joël	Professeur, UMR ESPACE-CNRS 6012, UAPV	Co-directeur
M. Feillet Dominique	Maître de conférences (HDR), LIA, UAPV	Examineur
M. Josselin Didier	Chargé de recherches, UMR ESPACE-CNRS 6012, UAPV	Examineur



École Doctorale 379 Temps, Espace, Pouvoir et Culture
UMR ESPACE-CNRS 6012, LIA

Résumés des chapitres de la thèse

1. Dans ce chapitre introductif, nous présentons les systèmes de transport à la demande (TAD) dans leur fonctionnement ainsi que la place qu'ils occupent dans le monde du transport de personnes. Les TAD sont abordés avec une vision élargie, en insistant sur les enjeux sociaux liés à l'évolution de la mobilité et les enjeux technologiques ou pratiques liés à la mise en place et la pérennisation de tels systèmes. Nous évoquons également les thèmes de recherche liés aux TAD qui peuvent susciter l'intérêt des communautés de la géographie et de la recherche opérationnelle.
2. Nous présentons un état de l'art sur les problèmes de calcul de tournées de véhicules principalement axé sur les problèmes de type ramassages et livraisons, et plus particulièrement liés au transport de personnes. Un intérêt particulier est porté aux méthodes de résolution exactes. Une introduction à une de ces méthodes est l'objet des sections 2.2 et 2.3. Il s'agit de fournir les bases pour appréhender la méthode de génération de colonnes qui décompose le problème en un problème maître, un programme linéaire généralement résolu à l'aide d'un solveur, et un problème esclave qui dans notre cas particulier prend la forme d'un Problème de Plus Court Chemin avec Contraintes de Ressources. Un algorithme de programmation dynamique générique résolvant le problème esclave est présenté. Il est utilisé, différemment adapté, tout au long de la thèse.
3. Après avoir défini formellement le problème étudié dans la section 3.1, nous proposons dans la section 3.2 une liste de critères de qualité de service, organisée suivant le niveau à partir duquel ils sont mesurables : à partir d'une longue période de fonctionnement, d'une solution complète, d'une tournée, de la course d'un passager ou d'un simple tronçon de route. Nous donnons une modélisation formelle de ces critères lorsque cela nous semble pouvoir rentrer dans le cadre de notre étude plus concentrée sur les critères mesurables à une petite échelle.
Nous considérons dans la section 3.3 l'optimisation des critères mesurables au moins sur une tournée, dans le cas où un véhicule connaît exactement les lieux qu'il doit visiter ainsi que l'ordre dans lequel ils doivent être visités. Le problème se restreint alors à un problème d'horodatage de la tournée, et/ou de choix des itinéraires à emprunter entre chaque lieux consécutifs.
4. Dans la section 4.1, nous proposons une méthode de génération de colonnes pour le problème de maximisation de la qualité de service présenté dans la section 3.1. Nous présentons les détails d'implémentation de la résolution des problèmes

maître et esclave, ainsi que les résultats obtenus sur des instances de PDPTW de la littérature. Dans la section 4.2, nous intégrons à ce schéma trois critères de qualité de service sélectionnés pour leur intérêt applicatif et discutons des modifications à apporter à la méthode générale pour les prendre en compte. Ces trois critères sont la distance totale parcourue, le temps perdu en transport et le taux de remplissage des véhicules. L'optimisation suivant ces trois critères est évaluée sur des instances dérivées de celle de PDPTW de la littérature. Hormis l'intégration des critères de qualité de service, un des objectifs de ce chapitre est de calculer et de comparer les solutions obtenues en optimisant suivant différents critères. Afin que les résultats obtenus, en terme de qualité des tournées soient comparables et ne soient pas imputables à la méthode de résolution, une résolution exacte s'impose.

5. L'application produite pour le TAD du Doubs Central est le sujet de ce chapitre. Après un bref aperçu des besoins et de la solution proposée (section 5.1), nous présentons la partie d'optimisation interne à l'application, dans la section 5.2. Le noyau de calcul des tournées est une heuristique d'insertion. Nous comparons ses performances avec une adaptation de la méthode de génération de colonnes du chapitre 4.
6. Après avoir mis en place des outils permettant la visualisation et la manipulation des tournées de véhicules dans leur environnement géographique, nous générons des instances réalistes correspondant à différents scénarios de demandes de transport. Nous comparons ensuite les solutions obtenues à partir des trois critères de qualité de service que sont la distance totale parcourue, le temps perdu et le taux de remplissage. Pour finir dans la section 6.2, nous étudions comment évaluer la sinuosité des tournées, vue comme un critère de qualité de service lié à la forme des tournées.

Table des matières

I	Transport à la demande	7
1	Enjeux, problématiques, méthodes	9
1.1	Évolution des besoins de mobilité et TAD	9
1.2	Mise en place d'un TAD et liens avec la Recherche Opérationnelle	16
2	Optimisation du calcul des tournées de véhicules : « Dial-A-Ride Problem »	21
2.1	État de l'art	21
2.1.1	Problèmes de calcul de tournées de véhicules	21
2.1.2	DARP	24
2.2	Méthode de génération de colonnes	31
2.2.1	Principes de la méthode	32
2.2.2	Contexte d'utilisation – Calcul de bornes inférieures pour des programmes linéaires en nombres entiers	34
2.2.3	Paramétrages – Efficacité	35
2.3	Plus court chemin avec contraintes de ressources (SPPRC)	36
2.3.1	Modélisation d'un ESPPRC	38
2.3.2	Résolution par programmation dynamique	39
II	Qualité de service pour le transport à la demande	43
3	Critères de qualité de service	45
3.1	Définition du problème étudié	45
3.2	Critères de qualité de service	47
3.2.1	Évaluation sur un tronçon ou un arrêt	49
3.2.2	Évaluation sur une course	52
3.2.3	Évaluation sur une tournée	54
3.2.4	Évaluation sur une solution	55
3.2.5	Évaluation à long terme	60
3.3	Maximisation de la qualité de service pour une séquence fixée d'arrêts à visiter	61
3.3.1	Définition et modélisation	61
3.3.2	Cas du 1-graphe	61
3.3.3	Cas du p -graphe	66

4	Maximisation de la qualité de service	71
4.1	Méthode de résolution par génération de colonnes	71
4.1.1	Modélisation	71
4.1.2	Adaptation de la méthode générale de génération de colonnes . .	75
4.1.3	Adaptation de la résolution du problème esclave	78
4.1.4	Résultats sur des problèmes standard de la littérature	89
4.2	Intégration des critères de qualité de service	99
4.2.1	La distance totale parcourue	99
4.2.2	Le temps perdu	102
4.2.3	Le taux de remplissage	106
4.2.4	Résultats	113
III	Étude de cas : Le TAD du Pays du Doubs Central	125
5	L'application développée	127
5.1	TAD à mettre en place	127
5.1.1	Description générale du TAD à réaliser	127
5.1.2	Définition du problème de calcul de tournées	129
5.1.3	L'application TADOU	132
5.2	Heuristique d'optimisation	134
5.2.1	Résolution par un algorithme d'insertion	134
5.2.2	Résultats	139
6	Visualisation des tournées	143
6.1	Plate-forme de travail couplée à un SIG	143
6.1.1	Structure de la plateforme Opensource	144
6.1.2	Construction d'instances typées	144
6.1.3	Critères de qualité de service et modèles de flux	148
6.2	Étude de la sinuosité	150
6.2.1	Définition de quatre critères d'évaluation	151
6.2.2	Observations	154
	Conclusion	165
	Liste des illustrations	165
	Liste des tableaux	167
	Bibliographie	169

Première partie

Transport à la demande

Chapitre 1

Enjeux, problématiques, méthodes

RÉSUMÉ : *Dans ce chapitre introductif, nous présentons les systèmes de transport à la demande (TAD) dans leur fonctionnement ainsi que la place qu'ils occupent dans le monde du transport de personnes. Les TAD sont abordés avec une vision élargie, en insistant sur les enjeux sociaux liés à l'évolution de la mobilité et les enjeux technologiques ou pratiques liés à la mise en place et la pérennisation de tels systèmes. Nous évoquons également les thèmes de recherche liés aux TAD qui peuvent susciter l'intérêt des communautés de la géographie et de la recherche opérationnelle.*

1.1 Évolution des besoins de mobilité et TAD

Les transports évoluent pour permettre d'aller plus loin, plus vite à un plus grand nombre d'individus. La quantité, la diversité et la longueur des flux de passagers augmentent. Ces dernières décennies, les transports ont pris de plus en plus de place dans notre quotidien. Le budget temps de chaque individu consacré au transport, longtemps constant à 12%, est passé à 20% (constat fait entre 1994 et 2000 en Suisse); ce qui à première vue contredit la loi de Zahavi (Zahavi et Ryan, 1980; Zahavi et Talvitie, 1980) sur le budget temps (et argent) constant. La journée « moyenne » a subi une désynchronisation temporelle avec par exemple une forte atténuation des pics de trafic, due entre autres à une souplesse de l'organisation du temps de travail accrue (Bailly et Heurgon, 2001). Qu'en est-il de notre mobilité ? et des enjeux économiques, politiques, sociaux, écologiques liés à la gestion de ces flux ?

De nombreuses disciplines s'intéressent à ces sujets. Une partie des géographes étudie l'interaction entre les réseaux de communication (transport ou autre) et les territoires, pour savoir dans quelle mesure les territoires forment les réseaux ou à quel niveau les réseaux (dé)structurent les territoires. Les sociologues s'interrogent sur l'impact d'une plus grande offre de mobilité dans notre société et sa fluidité. Est-ce que cette offre de quasi-ubiquité est égalitaire et génère un gain de liberté ? Ou bien, la société se voit-elle imposer de nouvelles contraintes dans un système élitiste séparant des

autres ceux qui ont un accès facilité à la mobilité ? Ils s'interrogent aussi sur les sources d'un tel besoin de mobilité ; s'il répond à un désir naturel de mouvement ou à des contraintes sociétales, comme par l'exemple la présence des habitats les moins onéreux loin des centres d'activités. Les industriels cherchent à proposer des produits basés sur de nouvelles technologies (GPS, Bluetooth, WI-FI...) satisfaisant ou anticipant les besoins en mobilité. Les transporteurs rationalisent leur offre de transport et leur gestion des flux via des outils d'optimisation et d'aide à la décision généralement basés sur des développements informatiques. En effet, lorsqu'un système de transport ne peut plus être géré manuellement, notamment lorsque la quantité de demandes de mobilité autorise potentiellement le regroupement des passagers dans les véhicules, l'utilisation de systèmes automatisés pour la recherche de solutions optimisées s'impose. Ces systèmes permettent de gérer des services au fonctionnement complexe soumis à de nombreuses contraintes. La recherche opérationnelle apporte ainsi une vision complémentaire à celle de l'expert de terrain, grâce à l'introduction de théories et de concepts éprouvés tels que la recherche du plus court chemin basée sur la théorie des graphes, les méthodes d'ordonnancement ou de programmation linéaire sous contraintes. Ces approches peuvent intervenir pour aider au choix des services (simulation préalable de fonctionnement du système), pour gérer le TAD (optimisation des tournées effectives) ou pour suivre son fonctionnement (audit et retour sur expérience). Le traité IGAT rédigé sous la direction de G. Finke (Finke, 2002) fournit un aperçu de ces différents aspects.

Dans cette thèse, nous nous intéressons au transport collectif individualisé correspondant aux exigences modernes et plus particulièrement au calcul et à l'optimisation des itinéraires via différents critères de qualité de service, à l'aide de méthodes adaptées provenant de la recherche opérationnelle.

Les individus recherchent des services de transport toujours plus souples, plus proches de leurs besoins. Malgré de récents efforts, les transports publics ne répondent que partiellement à ces attentes. Si ceux-ci restent compétitifs sur certains segments (en milieu urbain dense sur sites propres), dans la majorité des cas la voiture personnelle reste la plus à même de répondre aux besoins de déplacements individuels (Wiel, 2002; Dupuy, 1999). Toutefois, sur ces segments où les véhicules privés sont les plus compétitifs (zones péri-urbaines, rurales...), de par leur disponibilité et leur efficacité, des solutions alternatives existent (Castex et Josselin, 2007).

Relativement mal connus du grand public et plutôt négligés par les transporteurs, les transports à la demande (TAD), datant des années 1970, font leur réapparition depuis la fin des années 1990, à la faveur de l'avènement des technologies de l'information et de la communication (Ambrosino et Nelson, 2004) et de volontés politiques. Les TAD peuvent se définir comme : des « transports terrestres collectifs individualisés de personnes, activés seulement à la demande ». On les trouve actuellement essentiellement dans les pays riches (États-Unis, Europe, Japon). Les figures 1.1 et 1.2 illustrent le principe de fonctionnement d'un TAD.

La figure 1.1 représente les requêtes (flèches), les temps de trajet (pondération des flèches) estimés de l'origine (cercle surmonté de personnages) vers la destination. L'heure

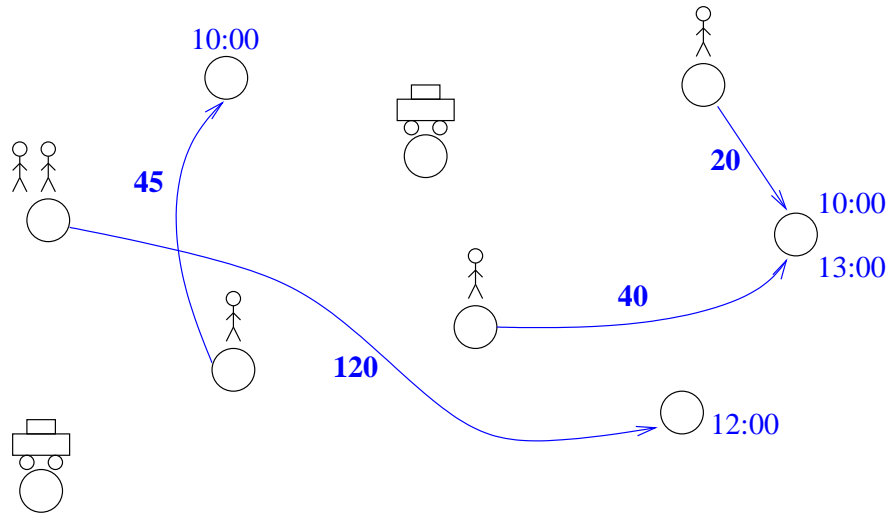


FIG. 1.1: Représentation du Transport à la Demande - Les demandes

d'arrivée souhaitée est indiquée à côté de la destination. Les cercles accolés à un véhicule représentent les dépôts des véhicules.

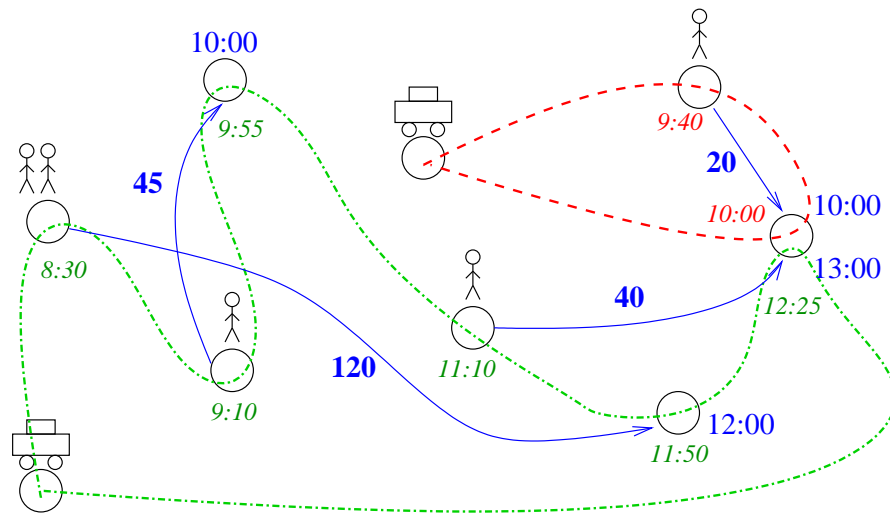


FIG. 1.2: Représentation du Transport à la Demande - Les Tournées

La figure 1.2 représente des tournées de véhicules répondant à la demande. L'itinéraire de chaque véhicule est représenté en pointillés. L'heure de passage est indiquée pour chaque lieu (ramassage ou livraison).

Historiquement, si on excepte le taxi-brousse africain et le taxi collectif italien, ce mode de transport prend ses origines aux États-Unis et commence à y être envisagé comme un mode de transport public. Les TAD possèdent différentes définitions et appellations qui correspondent à différentes philosophies allant du mode de transport palliatif – lorsqu'on ne peut pas faire autrement, par exemple dans des zones enclavées

– à une véritable révolution dans le monde des transports en commun, remplaçant alors les traditionnels taxis et lignes de bus ou de tramways comme l’imagine le concept de Modulobus (Josselin et Genre-Grandpierre, 2005). En 1966, les Américains votent un amendement sur le transport public qui débouche notamment sur un congrès en 1968, intitulé « Les transports de demain : de nouveaux systèmes pour les cités du futur »¹. C’est en 1970, que onze projets régionaux sont financés et permettent les premières études et expériences d’envergure. Ces projets sont principalement axés sur des aspects opérationnels et les informaticiens furent logiquement les chercheurs les plus sollicités, les autres disciplines ne s’emparant véritable du phénomène que bien plus tardivement. Le manque d’études sur l’intégration des TAD dans la société peut expliquer en partie la perte de vitesse du développement des TAD dans les années 80.

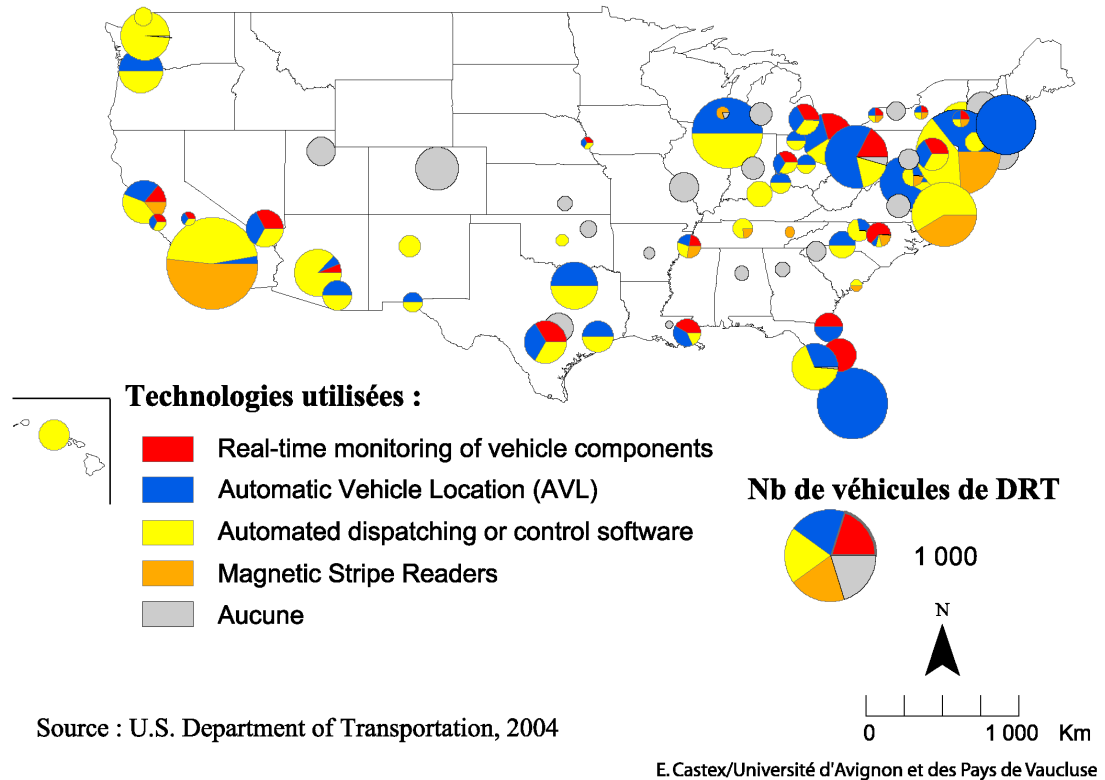
Les premiers TAD mis en place n’ont pas de vocation généraliste et concernent des niches d’usagers très restreintes, dont la plus représentative, encore à l’heure actuelle, est le transport de personnes à mobilité réduite (PMR) ou handicapées. Ce fut un des principaux déclencheur de la relance des recherches dans le domaine en 1990, suite à une loi américaine sur le développement des services aux PMR (« The American with Disabilities Act »). D’ailleurs, une des premières dénominations répandue pour ces systèmes est « *Paratransit Systems* ». Afin d’élargir l’image de ces systèmes de transport et d’insister sur leur caractère individualisé et parfois en temps réel, apparaît le terme de « *Dial-A-Ride* », peu à peu abandonné au profit d’appellations plus générales comme « *Demand Responsive Transport* » et « *Transport On-Demand* » (ou « *On-Demand Transportation* ») suivant les communautés scientifiques. En France comme en Europe, les TAD se développent au gré des législateurs qui leur donnent un cadre juridique et surtout des crédits. En France, la Loi d’Orientation des Transports Intérieurs (LOTI)² les définit juridiquement comme « des services collectifs offerts à la place, déterminés en fonction de la demande des usagers, dont les règles générales de tarification sont établies à l’avance et qui sont exécutés avec des véhicules dont la capacité minimale est fixée par décret ». Ces systèmes ont une forte connotation de transport public et sont donc axés sur la demande, d’où le terme de « Transport À la Demande » et même les appellations de « (Mini)Bus À la Demande » permettant de les distinguer nettement des systèmes de co-voiturage ou de taxis collectifs (« *car-pooling* » et « *car-sharing* » en anglais) avec lesquels ils sont souvent assimilés par le grand public. Par la suite, nous emploierons la formulation « Transport À la Demande » (TAD) de façon générique dans le sens donné par Ascher (2002)(page 278) : « *Le transport à la demande est une notion générique qui englobe a priori tous les services de transport dont tout ou partie ne s’effectue qu’à la demande expresse de ceux qui les utilisent* ».

Les TAD tardent à se développer en France, et les États-Unis restent largement le pays le plus riche en TAD dans ses zones les plus fortement urbanisées, comme le montre la carte de la figure 1.3 recensant les systèmes de transports intelligents (*Intelligent Transport Systems (ITS)* en anglais) aux États-Unis. Leur implantation n’a cessé d’augmenter depuis les années 90 passant de 42,4 millions de voyages en 1991 à 73,2

¹De l’américain : « *Tomorrow’s transportation : new systems for the urban futur* ».

²La LOTI fut votée en 1982 et suivie d’un décret en 1985, puis précisée par la Loi sur l’Air et l’Utilisation Rationnelle de l’Énergie (loi LAURE ou Lepage) du 30 décembre 1996.

millions en 2000. En France, plus de 600 services de TAD existent, couvrant plus de



Traduction des noms des technologies utilisées en français pour les TAD (=DRT)

Real time monitoring of vehicle components : Gestion des véhicules en temps réel.
Automatic Vehicle Location : Localisation de véhicule de type GPS.
Automated dispatching or control software : Logiciels de gestion.
Magnetic Stripe Readers : Lecteurs de bandes magnétiques.

FIG. 1.3: Carte des Systèmes de Transport Intelligents aux États-Unis par type de technologie.

6000 communes. Ces services sont souvent développés pour maîtriser les dépenses de transport (remplacement des lignes de bus déficitaires), assurer le droit à la mobilité à certaines populations, captives, défavorisées (PMR, par exemple) ou proposer un mode de déplacement adapté aux zones de faibles densités, aux heures de frange de la journée (matinées, soirées, nuits) ou aux évènements localisés, occasionnels ou récurrents. Un dernier argument est celui de la lutte contre la pollution, puisque l'optimisation des tournées et le regroupement des voyageurs permettent en effet une économie potentiellement importante de véhicules roulant et donc d'émission de polluants atmosphériques (Houzet et Josselin, 2004). Dans sa thèse, Castex (2007) dresse un état des lieux des TAD en France. Il en ressort que les TAD sont souvent mis en place pour palier un déficit des transports en commun. Leur fonctionnement dépend fortement de la nature du service à rendre et des moyens financiers et techniques disponibles. Ainsi, à l'heure actuelle 70% des TAD français opérationnels sont très faiblement flexibles et jouent sur

des ajustements de lignes fixes au niveau des horaires ou des ordres de passage. Parmi les autres TAD qui mettent en œuvre des systèmes plus évolués de prise en compte de la demande, comme des outils informatiques de gestion et d'optimisation, moins de la moitié ont une gestion dynamique de la demande. Cependant, les TAD dynamiques sont un objectif idéal de fonctionnement pour une grande partie des TAD. Les TAD dynamiques représentent donc une part importante des travaux de recherche sur le sujet.

Dans le tableau 1.1, nous présentons les avantages (marqués par un plus) et les inconvénients (marqués par un moins) de différents modes de transport – la voiture particulière (VP), le bus ou tramway ou métro, le taxi traditionnel, la bicyclette ou la marche à pied et le TAD, par rapport à des critères d'exigences exprimables par les différents acteurs du transport. Bien sûr, on peut supposer que les aspirations des usagers sont incluses dans celles des Autorités Organisatrices de Transport (AOT, prescripteurs relayant la demande sociale via les élus) qui souhaitent le bien-être des citoyens, ainsi que celles des AOT le sont dans celles des transporteurs qui doivent rendre des comptes aux AOT.

	critère	VP	BUS	TAXI	VÉLO	TAD
U	économie	- cher	+ bon marché	- très cher	+ bon marché	+ bon marché
S	rapidité	+ pas d'arrêts	+ voies réservées	+ voies réservées	- trop physique	+ voies réservées
A	choix des horaires	- subit le trafic	- arrêts longs	+ pas d'arrêts	+ libre	+ peu d'arrêts
G		+ libre	- horaires fixes	+ réservation		+ réservation
E	précision	- stationnement	- arrêts fixes	+ porte à porte	+ porte à porte	+ porte à porte
R	agrément	- conduite	- promiscuité	+ confort	- effort	+ confort
S	sécurité	- dans le trafic	+ chauffeur pro	+ chauffeur pro	- fragilité/vol	+ chauffeur pro
A	économie	- contrôle routier	- non modulable	+ privé	+ privé	+ adapté à la demande
O	égalité	- cher	- dessertes limitées	- cher	- sportif	+ accessible à tous
T	écologie	- individuel	+ regroupement	- individuel	+ non polluant	+ regroupement
S**	fluide	- individuel	+ regroupement	- individuel	+ peu encombrant	+ regroupement
C	économique	- non concerné	- investissements	+ forte rentabilité	non concerné	+ modulable
T*	gestion	+ non concerné	+ régulière	+ en direct	+ non concerné	- complexe

(*) Autorités Organisatrices du transport

(*) CT : Compagnies de transporteur

TAB. 1.1: Comparatif entre différents modes de transport

1.2 Mise en place d'un TAD et liens avec la Recherche Opérationnelle

Les TAD mettent en scène autour des besoins de mobilité quatre acteurs : les AOT, les transporteurs, les passagers et le territoire incluant le réseau de communication. De la diversité de ces éléments, naît celle des offres de transport, ce qui rend la classification des TAD assez ardue. Dans sa thèse de géographie, [Castex \(2007\)](#) propose une analyse par correspondances multiples, une analyse par composantes principales et une analyse factorielle sur une base de données contenant l'ensemble des TAD français recensés à ce jour et décrits par une trentaine d'indicateurs. De ces analyses, ressortent nettement les distinctions entre les TAD urbains, ruraux et péri-urbains. Le territoire semble donc avoir une forte influence sur le type d'offre avec une tendance à une plus grande flexibilité en milieu urbain. Nous évoquons ici simplement les caractéristiques de l'offre de transport qui nous semblent essentielles.

Entre le taxi qui fait du porte à porte et le bus qui suit des lignes fixes, le TAD peut, en plus, respecter d'autres types de trajets comme d'arrêt à arrêt, ou des lignes fixes à déclenchement (suite à au moins une demande). Les courses des véhicules peuvent être plus ou moins polarisées et peuvent aller jusqu'à une convergence (ou divergence) stricte, par exemple pour des événements particuliers comme l'arrivée d'un train. À cela s'ajoute l'inter-modalité du transport déjà très développée par les services de livraisons mais pour laquelle, nous ne connaissons pas encore d'application opérationnelle pour le transport de passagers à la demande.

Le mode de réservation peut être plus ou moins contraignant pour les usagers, la gestion en temps réel avec des délais d'attente, et une quantité de rejets faibles est souvent un objectif pour les transporteurs.

Une troisième caractéristique nous paraissant essentielle, est la souplesse demandée au client en terme de détour ou d'horaire, ce qui représente souvent le critère principal en terme de qualité de service. Ceci peut être relié au mode de tarification qui a un impact majeur sur la fréquentation et le mode d'utilisation du service.

Tout système de TAD a deux vies ; une à l'état de projet et une en tant que système opérationnel. La principale difficulté pour la gestion du projet est celle de ses acteurs. Les premiers acteurs à considérer sont les différentes AOT compétentes, qui, suivant le type de réseau ou le territoire changent et ne sont pas toujours prêts à coopérer dans le même sens. La figure 1.4 indique les secteurs de compétence des principales AOT françaises. Bien sûr, la cible des usagers doit être parfaitement définie afin d'établir un système adéquat en termes de service et de tarification. Enfin, les transporteurs locaux qu'ils soient partenaires ou concurrents imposent des contraintes assez fortes sur l'offre de transport pouvant être émise.

D'un point de vue opérationnel, un premier problème est de définir un système de transport optimal en termes de types de véhicules, de localisation des dépôts et de tarification. L'aspect volumineux et parfois flou des contraintes formulées lors de la phase de projet, ne permettent souvent pas de formaliser mathématiquement le pro-

1.2. Mise en place d'un TAD et liens avec la Recherche Opérationnelle

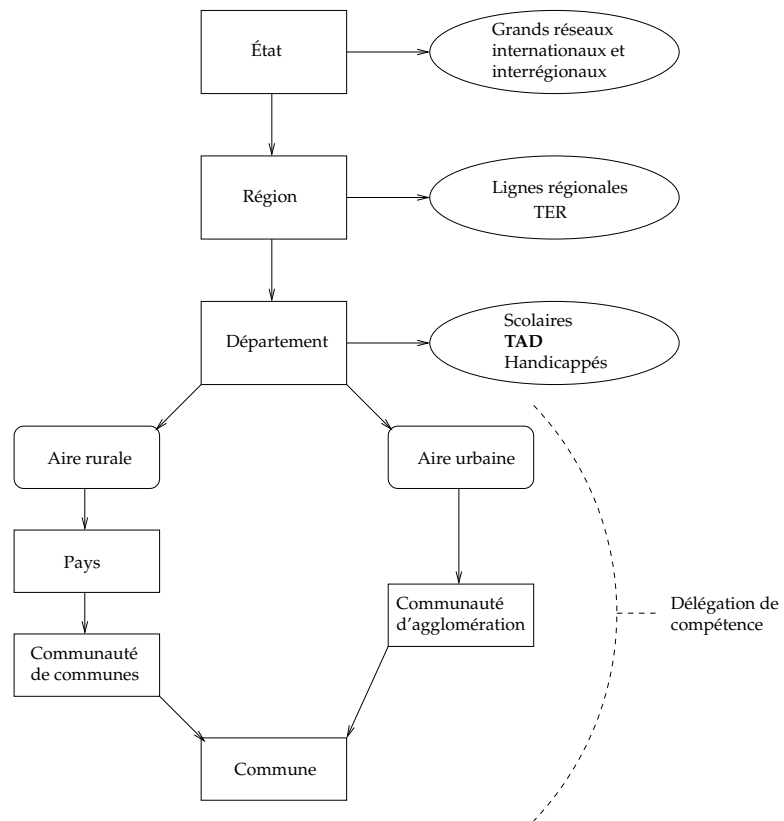


FIG. 1.4: Niveaux de compétence des AOT en France

blème pour le résoudre. Cependant, la géostatique spatiale permet de mettre en relation des structures spatiales de demande affinées (semis de population, taux de pénétration, matrices origines/destinations) sans lesquelles aucun système de transport opérationnel ne pourrait être efficacement mis en place. Les méthodes de géosimulation sont ainsi très développées en géographie. Celles-ci peuvent s'appuyer sur des sous-problèmes plus académiques tel que le problème de localisation (Drezner, 1996), pour fixer les lieux des dépôts d'après une estimation des flux de passagers, et le problème de calcul de tournées de véhicules établissant les feuilles de route des véhicules pour une liste de flux, une fois les dépôts fixés. Des travaux récents permettent même d'envisager le traitement conjoint de ces deux problématiques, localisation et routage (Prins et al., 2006, 2007). Ces deux problématiques font appel à des méthodes relevant de la recherche opérationnelle. Au stade de la simulation, les données fournies sont le fruit direct d'études terrain (enquêtes, historiques...) ou d'études analytiques (structures des réseaux, du territoire, des semis de population...). Différents types de méthodes de résolution peuvent être envisagées en fonction du temps imparti au calcul, du volume de données et du degré d'exigence en terme de qualité des solutions. Ces paramètres guident vers des approches *heuristiques* pour de grands volumes de données et des réponses rapides, d'*approximation* pour des réponses rapides avec une garantie sur la qualité des solutions ou *exactes* nécessitant plus de temps de calcul et souvent envisageables sur des petits volumes de données.

Nos travaux se restreignent à l'étude des tournées de véhicules dont voici une brève présentation. Dans son fonctionnement, un TAD permet de répondre à des demandes (ou « requêtes ») de transport via une flotte de véhicules sous certaines contraintes de faisabilité et de fonctionnement. Les TAD posèrent, dès leur apparition, des problèmes d'optimisation de construction des tournées et d'aide à la décision nouveaux par rapport aux problèmes de transport de passagers classiques où la planification des tournées n'est pas quotidienne. Les enjeux de tels systèmes consistent à trouver des solutions optimisées de fonctionnement associant les objectifs, parfois contradictoires, des partenaires impliqués :

- rationaliser l'offre et rendre visible et attractif le service pour les AOT,
- rentabiliser le service pour les transporteurs (sociétés auxquelles les AOT délèguent l'organisation des transports sur leur territoire),
- conquérir de nouveaux marchés pour les sous-traitants éventuels (taxis),
- améliorer le cadre de vie et l'accès aux facilités (associations d'usagers).

Ces objectifs impliquent la prise en compte de critères économiques (coût de fonctionnement), sociaux (accessibilité, équité, coûts attractifs, convivialité), fonctionnels (réactivité, flexibilité, niveau de remplissage des véhicules), environnementaux (pollution, fluidité du trafic urbain, nuisances sonores et encombrement des parkings en zones urbaines).

Cette thèse propose des méthodes pour le calcul de tournées de véhicules avec les objectifs de prendre en compte différents critères de qualité de service et de comparer les tendances observées sur les solutions obtenues. Afin de s'affranchir du biais pouvant être généré sur la comparaison des solutions par l'utilisation de méthodes inexactes, nous utilisons des méthodes de résolution exacte. Un projet de recherche-action

1.2. Mise en place d'un TAD et liens avec la Recherche Opérationnelle

effectif dans le Pays du Doubs Central nous permet d'expérimenter certains de nos travaux.

Chapitre 2

Optimisation du calcul des tournées de véhicules : « Dial-A-Ride Problem »

RÉSUMÉ : Nous présentons un état de l'art sur les problèmes de calcul de tournées de véhicules principalement axé sur les problèmes de type ramassages et livraisons, et plus particulièrement liés au transport de personnes. Un intérêt particulier est porté aux méthodes de résolution exactes. Une introduction à une de ces méthodes est l'objet des sections 2.2 et 2.3. Il s'agit de fournir les bases pour appréhender la méthode de génération de colonnes qui décompose le problème en un problème maître, un programme linéaire généralement résolu à l'aide d'un solveur, et un problème esclave qui dans notre cas particulier prend la forme d'un Problème de Plus Court Chemin avec Contraintes de Ressources. Un algorithme de programmation dynamique générique résolvant le problème esclave est présenté. Il est utilisé, différemment adapté, tout au long de la thèse.

2.1 État de l'art

2.1.1 Problèmes de calcul de tournées de véhicules

2.1.1.1 Terminologie

Le système de transport est défini par une *flotte de véhicules* basés dans des *dépôts*. Chaque véhicule peut avoir ses propres caractéristiques (capacité, équipements, vitesse de déplacement, coût d'utilisation, durée d'utilisation maximale...).

Une *requête* est une demande de *services*, un ramassage et une livraison en général, formulée par des passagers.

Un *tronçon* est un itinéraire continu (sans arrêts) entre deux services et identifiable sur le réseau routier.

Une *tournée* est le plan de route exact – tronçons empruntés et heures de passages comprises si nécessaire – attribuable à un véhicule.

Une *course* est un fragment de tournée relatif à une requête ; c'est-à-dire ce qui sépare le ramassage initial de la livraison finale.

Une *séquence* est une liste ordonnée de services à effectuer pour un véhicule ; les tronçons empruntés n'étant pas forcément déterminés.

Une contrainte de *fenêtre de temps* correspond à un couple de dates de réalisation au plus tôt et au plus tard, attribué à un service.

Les procédures d'optimisation par séparation et évaluation progressive (*Branch & Bound* en anglais) sont désignées sous l'acronyme SEP. On précise les cas où une génération de coupes y est incluse par SEPGCOU (*Branch & Cut* en anglais), ou une génération de colonnes par SEPGCOL (*Branch & Price* en anglais), ou bien les deux par SEPGCC (*Branch & Cut & Price* en anglais).

2.1.1.2 Le TAD parmi les problèmes de transport

Une fois le système de transport défini, son exploitation nécessite généralement une procédure automatisée du calcul des tournées des véhicules. Les problèmes de transport représentent une thématique majeure en recherche opérationnelle, tant quantitativement qu'historiquement.

Le problème de calcul de tournées de véhicules sous-jacent aux problèmes de TAD est connu sous la dénomination générique de *Dial-A-Ride Problem* (DARP). D'un point de vue formel, ce problème apparaît comme une suite d'extensions de problèmes de transport dont nous présentons les caractéristiques respectives dans le tableau 2.1.

problème	dépôt	véhicules	capacité	fenêtres de temps	couplage	autre
TSP						
mTSP	×	×				
CVRP	×	×	×			
VRPTW	×	×	×	×		
PDP	×	×	×		×	
PDPTW	×	×	×	×	×	
DARP	×	×	×	×	×	×

TAB. 2.1: Récapitulatif des caractéristiques de quelques problèmes classiques

Le célèbre *Problème du Voyageur de Commerce*, ou *Traveling Salesman Problem* (TSP) proche du problème de recherche d'un circuit hamiltonien, est connu des mathématiciens et des statisticiens depuis le 19^e siècle. Dans sa formulation aujourd'hui classique, il s'agit de trouver l'ordre de parcours de villes à visiter par un commerçant itinérant, en

minimisant la distance totale parcourue. On attribue à Karl Menger la première publication à propos de la résolution du TSP (Menger, 1932), où il en considère une variation qu'il appelle *das Botenproblem* ou *the Messenger Problem*. La résolution de ce problème sous sa forme originale resta un challenge très concurrentiel en optimisation combinatoire pendant plusieurs décennies. Dans les années 90 David Applegate, Robert Bixby, Vasek Chvátal et William Cook développèrent la méthode Concorde¹, basée sur une SEPGCOU qui permet de résoudre optimalement des instances de très grande taille (par exemple le TSP de toutes les villes de Suède).

Des nombreux travaux traitant du TSP et de ses variantes, sont issues bon nombre de méthodes classiques en optimisation combinatoire et recherche opérationnelle et plus particulièrement dans le domaine de la résolution des programmes linéaires en nombres entiers. Ces méthodes sont les *Plans de Coupe* (ou *Cutting Planes*) de Dantzig et al. (1954), le SEP avec l'étude du cas asymétrique (Eastman, 1958) puis généralisée par Land et Doig (1960). Les chercheurs abordèrent également ces problèmes de façon heuristique, comme par exemple avec de la recherche locale et le toujours très utilisé opérateur 2-Opt (Croes, 1958). Les progrès effectués en terme de résolution et la variété des applications aux systèmes de transport amenèrent les chercheurs à considérer de nombreuses extensions ou variantes du TSP.

Hormis le cas asymétrique (ATSP), une des premières extensions naturelles du TSP fut de considérer le transport de marchandises récoltées en cours de trajet et ramenées à un dépôt unique par plus d'un véhicule, c'est le problème standard de calcul de tournées de véhicules ou *Vehicle Routing Problem* (VRP). Ce problème prend tout son intérêt si on y intègre une contrainte de capacité sur les véhicules pour le *Capacited Vehicle Routing Problem* (CVRP). Les problèmes se sont ensuite enrichis par des contraintes temporelles (fenêtres de temps) sur les lieux à visiter pour le *Vehicle Routing Problem with Time Windows* (VRPTW) ; des contraintes de précédence entre les lieux à visiter par exemple dans le *Problème de Ramassages et Livraisons* ou *Pick Up & Delivery Problem* (PDP), où les marchandises doivent être transportées d'un lieu vers un autre ; puis d'autres contraintes de qualités de service de plus en plus fines, comme le temps de transport maximal, liées au transport de biens particuliers et notamment au transport de personnes. Ces cas particuliers de PDP sont souvent désignés sous le terme de *Dial-a-Ride Problems* (DARP), ou *Problèmes de Transport à la Demande* dont un état de l'art détaillé fait l'objet de la section 2.1.2. Dans une majeure partie de la littérature, l'objectif est la minimisation de la distance totale parcourue. Dans un DARP, l'intégration de certains critères de qualité de service n'ayant pas d'expression naturelle sous forme de contraintes, peut conduire à une fonction objectif différente.

D'autres extensions du TSP dans la littérature sont à relever, comme les problèmes avec fractionnement de la demande où les biens transportés peuvent être déposés puis rechargés et dont la version la plus simple est le *Split & Delivery Problem* dans lequel une visite en un sommet peut ne donner lieu qu'à un chargement partiel des biens disponibles. Une famille importante des problèmes de tournées de véhicules est celle considérant un routage sur les arcs et non sur les sommets, cette famille est à relier aux

¹<http://www-neos.mcs.anl.gov>

problèmes de recherche de circuits eulériens dans un graphe et dont le représentant standard est le *Problème du Postier Chinois*. Un temps délaissés, ces problèmes particuliers trouvent à l'heure actuelle un regain d'intérêt du au nombre important d'applications de cette nature, comme la collecte des déchets et le déenneigement ou le salage des voies (Assad et Golden, 1995). Ils deviennent ainsi un champ d'application ouvert aux méthodes d'optimisation comme les algorithmes génétiques (Lacomme et al., 2005). Les systèmes de transport inter-modaux, c'est-à-dire avec différents modes de transport, offrent aussi des problèmes beaucoup étudiés. L'optimisation de la localisation même des entrepôts est un sujet d'étude, comme dans le *Location Routing Problem* (Prins et al., 2006). Également, les problèmes de calcul de tournées de véhicules sont couplés avec la gestion des stocks, comme dans le *Inventory Routing Problem*, ou avec la gestion de l'emploi du temps du personnel. Nous conseillons au lecteur intéressé par les variantes des problèmes de tournées de véhicules les ouvrages suivants : Gutin et Punnen (2002); Toth et Vigo (2002); Crainic et Laporte (1998).

Une classification transversale aux problèmes de transport est basée sur la distinction entre problèmes statiques, pour lesquels toutes les données sont connues avant le calcul, les problèmes stochastiques, où les données est connue par une fonction de probabilité et les problèmes dynamiques, où des nouvelles données peuvent apparaître en cours de calcul. Les évolutions technologiques actuelles, notamment dans les Technologies de l'Information et de la Communication (TIC) comme le GPS, le Wimax et tant d'autres, et leur large diffusion rendent accessibles des applications proches du temps réel qui deviennent un axe de recherche attractif (Semet et Malca, 2006). Cependant, pour beaucoup de systèmes, une bonne partie des requêtes sont connues par avance. Un algorithme résolvant le cas statique est alors utile. La fonction objectif de ce dernier peut tenir compte de l'arrivée probable de futures requêtes, cadre de l'optimisation robuste.

2.1.2 DARP

2.1.2.1 Historique

Le calcul de tournées de véhicules dans les TAD commence à être traité comme un problème d'optimisation combinatoire dans les années 70. Les systèmes étudiés sont en temps réel (dynamiques) et orientés vers la qualité de service des passagers ou des chauffeurs. Des études analytiques sont développées pour prédire le comportement de tels systèmes soumis à des demandes respectant des règles probabilistes. Daganzo (1978) établit un modèle analytique pour la prévision des temps d'attente et des temps de transport comme fonctions du nombre de véhicules et de la quantité de demande. Trois algorithmes régissant l'affectation des demandes aux véhicules à partir de notions de proximité, sont testés puis comparés sur les critères du temps d'attente et du temps de transport. Daganzo (1984) compare le comportement d'un TAD de type porte-à-porte avec un système à lignes fixes (de type bus de ville). Le système porte-à-porte devient d'autant plus attractif que la quantité de demande diminue. Cependant d'après cette étude, le coût d'exploitation du système porte-à-porte est aussi plus intéressant.

Le modèle analytique développé par [Stein \(1978\)](#) mesure le temps de transport moyen pour un TAD, où les véhicules ont des zones d'attribution et où la demande est élevée et respecte une loi de Poisson. [Wilson et al.](#) étudient de manière intensive les TAD en temps réel ([Wilson et al., 1971](#); [Wilson et Weissberg, 1976](#); [Wilson et Colvin, 1976](#)). Ils proposent des heuristiques basées sur des principes de meilleure insertion qu'ils évaluent en fonction du temps d'attente, du temps de transport, des détours et de l'écart par rapport à une date d'arrivée souhaitée.

Dans les années 80, faute d'applications, l'engouement pour les TAD s'est quelque peu émoussé. En effet, les technologies de l'époque ont du mal à rendre les TAD aussi attractifs qu'espéré. Quelques travaux de références sont malgré tout publiés [Psarafitis \(1980, 1983a,b\)](#); [Sexton et Bodin \(1985a,b\)](#); [Bodin et Sexton \(1986\)](#); [Sexton et Choi \(1986\)](#) sur le problème à un véhicule, ou, [Jaw et al. \(1986\)](#) pour le problème à plusieurs véhicules.

Dans les années 90 un nouvel intérêt pour les TAD apparaît, lié aux progrès techniques de l'époque notamment dans le domaine des TIC. Jusqu'alors prévus pour des catégories restreintes de personnes (scolaires, handicapés...), les TAD sont envisagés à grande échelle (plusieurs milliers de personnes par jour) et comme systèmes de transport en commun ([Horn, 2002, 2003, 2004](#)). Les méthodes heuristiques sont alors nettement privilégiées et les critères d'optimisation deviennent des compromis entre les coûts d'exploitation et la qualité de service. Généralement, ces critères sont la distance totale parcourue, la durée d'une tournée, la durée de transport individuel, l'écart par rapport à une date de service souhaitée. Une tendance actuelle est d'essayer d'intégrer d'autres critères d'optimisation ; ce qui amène aussi à l'étude d'outils d'aide à la décision multicritères.

Certains DARP ont une structure vraiment particulière qui les rapproche d'un problème de VRPTW classique. Il s'agit de problèmes de rabattement, par exemple par des taxis pour des voyageurs vers un départ de train ([Chevrier et al., 2006a](#)) ou en covoiturage par des employés d'une même entreprise vers leur lieu de travail ([Baldacci et al., 2004](#)). Ce pôle de convergence (ou de divergence) est parfois assimilable à un dépôt de véhicules unique et les requêtes se simplifient en de simples ramassages (ou livraisons).

Soit pour des raisons de complexité, de volume de données, ou d'aspects dynamiques les TAD posent des problèmes généralement propices à des approches heuristiques qui forment la majorité de la littérature à ce sujet. La section suivante est consacrée à une rapide présentation des principales méthodes de résolution proposées jusqu'alors dans le domaine. Des états de l'art plus détaillés peuvent être trouvés dans [Desaulniers et al. \(2002\)](#) sur les problèmes de *Pick-Up & Delivery* et dans [Savelsbergh et Sol \(1995\)](#) sur leur extension en un problème de transport plus général : *The General Pick-Up & Delivery Problem*. Les articles de [Cordeau et Laporte \(2003a, 2008\)](#) et de [Cordeau et al. \(2004\)](#) sont dédiés aux DARP dérivant de PDPTW. Dans [Paquette et al. \(2007\)](#) on trouve une étude des différentes fonctions objectifs envisagées et l'intégration de quatre d'entre elles dans la recherche tabou de [Cordeau et Laporte \(2003a\)](#).

2.1.2.2 Approches de résolution

Le DARP est structurellement très proche du PDPTW, lui-même une extension du VRPTW. La différence se fait généralement au niveau de la détermination des dates de service et de la fonction objectif. De nombreuses méthodes résolvant le PDPTW sont des adaptations de celles résolvant le VRPTW. Il en va de même pour le DARP. Le PDPTW est un problème pour lequel prouver la réalisabilité est NP-complet, comme extension du VRPTW. L'adaptation des méthodes efficaces pour le VRPTW vers le PDPTW et ses extensions doit être effectuée avec attention. En effet, certaines contraintes peuvent changer la classe de complexité d'un calcul de voisinage (en recherche locale par exemple) ou d'un calcul de dominance (en programmation dynamique par exemple). Quelques cas ont déjà fait l'objet de travaux, comme tester la validité d'une séquence de services soumise à des contraintes de couplage et de précédence entre ramassages et livraisons, de temps d'attente et de temps de transport limités. [Hunsaker et Savelsbergh \(2002\)](#) prouvent que la vérification de la validité d'une telle séquence peut se faire en temps linéaire. [Ahuja et al. \(1998\)](#) détectent l'irréalisabilité de l'insertion d'une requête dans une séquence en temps constant avec une procédure de mise à jour en $\mathcal{O}(n^2 \log n)$.

Du caractère naturellement multicontraint du DARP, d'autres approches purement multicritères ou multiobjectifs commencent à être étudiées en vue de fournir des outils d'aide à la décision aux transporteurs.

Le DARP a une décomposition naturelle en trois sous-problèmes :

1. *Un problème d'affectation*, qui consiste à déterminer par quel véhicule est servie chaque requête ; ce problème est connu sous le nom de *Clustering Problem*.
2. *Un problème de séquençement*, qui consiste à déterminer dans quel ordre chaque véhicule visite les lieux qui lui sont attribués ; ce problème est connu sous le nom de *Routing Problem*.
3. *Un problème d'horodatage*, qui consiste à déterminer l'heure de passage des véhicules en chaque lieu ; ce problème est connu sous le nom de *Scheduling Problem*².

Ces trois sous-problèmes peuvent être traités globalement par coopération d'algorithmes, ou indépendamment par une séquence d'algorithmes. Des phases de post-optimisation via des recherches locales ou tabou sont souvent intégrées à la fin de chaque phase. Lors d'une décomposition *clustering first - routing second*, le problème de *routing* correspond à un DARP à un seul véhicule. Dans cette section, nous présentons ou citons des travaux réalisés dans le cadre d'études de DARP ou de problèmes proches. Nous nous consacrons presque exclusivement à des problèmes statiques, considérant comme seul mode de transport la voiture. Nous avons séparé les publications en trois groupes : les algorithmes d'approximation, les méthodes exactes et les heuristiques.

2.1.2.2.1 Algorithmes d'approximation La résolution de DARP par des algorithmes d'approximation est marginale car les systèmes de transport sont trop exigeants en

²On trouve également le terme d'ordonnancement, qui, en français regroupe le séquençement et l'horodatage.

terme de qualité moyenne des solutions sur plusieurs itérations et non en terme de pire cas. A notre connaissance, ce type d'approche a été appliqué uniquement sur des relaxations du DARP. Le 1-PDP (PDP à un seul véhicule) sans contrainte de capacité est résolu par [Hauptmeier et al. \(2001\)](#) avec des performances de $5/3$ lorsque le réseau est un arbre et de $9/4$ dans le cas général. [Charikar et Raghavachari \(1998\)](#) donnent un algorithme d'approximation de performance en $\min\{\mathcal{O}(\log N), \mathcal{O}(k)\}$ pour une version préemptive d'un 1-PDP avec contrainte de capacité ; [Gørtz \(2006\)](#) montre, pour le même problème, que pour tout $\epsilon > 0$ on ne peut pas atteindre une performance de $\min\{\mathcal{O}(\log^{1/4-\epsilon} N), k^{1-\epsilon}\}$.

2.1.2.2 Méthodes exactes La résolution du DARP par des méthodes exactes est rarement la priorité des applications. En effet, l'obtention d'une solution optimale n'est pas souvent jugée pertinente dans le cas dynamique qui représente une part importante des problèmes étudiés. Les premières méthodes exactes concernent des sous-problèmes du DARP. Dans le cas à un véhicule, [Psaraftis \(1980\)](#) résout par programmation dynamique un DARP à un véhicule en version statique et dynamique. La qualité de service offerte aux passagers est assurée par des contraintes originales MPS (*Maximum Position Shift*) qui garantissent une limite sur la position d'apparition dans la séquence pour chaque requête. Il adapte ensuite son algorithme à un problème comprenant des fenêtres de temps sur les dates de réalisation des services ([Psaraftis, 1983b](#)). La fonction objectif minimise la durée totale de la tournée, le temps d'attente des passagers aux arrêts et la durée passée par les passagers dans le véhicule. Les instances résolues ne dépassent pas une dizaine de passagers. Un autre algorithme de programmation dynamique ([Desrosiers et al., 1986](#)) permet de résoudre le 1-PDPTW (sans autres contraintes de qualité de service) efficacement sur des instances de plus grande taille (55 requêtes), à condition que la contrainte sur la capacité des véhicules soit assez restrictive. Même si la version à un seul véhicule n'apparaît pas dans les applications réelles, elle est souvent rencontrée comme sous-problème de méthodes générales exactes, heuristiques ou dynamiques. Notons que le même type d'algorithme de programmation dynamique a été appliqué avec succès aux TAD convergents par [Chevrier et al. \(2006a\)](#).

La première résolution exacte du PDPTW complet est due à [Dumas et al. \(1991\)](#) qui proposent un algorithme basé sur une SEPGCOL capable de fermer des instances à 40 requêtes dans le cadre de transport de personnes handicapées. Le schéma de résolution utilise une formulation en un problème d'affectation, où les colonnes sont des tournées valides et où les contraintes assurent le service unique de chaque requête. Le générateur de colonnes (problème esclave) résout un problème de plus court chemin sous contraintes de ressources. Hormis les fenêtres de temps, un niveau de qualité de service supplémentaire des passagers est pris en compte lors de l'horodatage des tournées en minimisant les temps d'attente des passagers. Les règles de branchement sont adaptées en fonction de la profondeur dans l'arbre de recherche. En haut de l'arbre, les branchements sont faits sur les véhicules puis les fenêtres de temps. En bas de l'arbre, la convergence est assurée par des branchements sur les successions (arcs) entre paires de services. Cette approche est assez robuste dans le sens où elle supporte bien le passage à une flotte de véhicules hétérogène et où de nombreuses contraintes applicatives

peuvent s'interpréter comme des ressources dans le problème esclave. [Savelsbergh et Sol \(1998\)](#) apportent quelques modifications à ce schéma. Ils gèrent de façon plus fine l'ensemble des colonnes traitées, afin d'éviter les redondances et de sélectionner des colonnes prometteuses en vue d'obtenir une solution entière. Le problème esclave est résolu de façon heuristique avant d'envisager la nécessité d'une résolution exacte. Les branchements se font sur des variables représentant la fraction de requête servie par un véhicule donné. Dernièrement, [Ropke et Cordeau \(Ropke, 2005\)](#) proposent un algorithme basé sur le même schéma pour résoudre le PDPTW. Leurs principales contributions sont l'ajout de différentes coupes et la résolution du problème esclave par différentes heuristiques avant d'utiliser une méthode exacte. Cette dernière résout le problème esclave dans sa version élémentaire, idée déjà utilisée par [Feillet et al. \(2004\)](#) et [Chabrier \(2006\)](#), ce qui augmente significativement la qualité de la relaxation linéaire du problème maître. La méthode est adaptée au DARP, c'est-à-dire qu'une limite sur le temps de transport des passagers est ajoutée. Cette contrainte n'est pas intégrée au problème esclave mais est gérée par des coupes. Les résultats sont très bons sur les deux problèmes (DARP et PDPTW); des instances entre 100 et 200 requêtes, suivant leur structure, sont fermées. Dans le cadre de transport d'animaux, [Sigurd et al. \(2004a\)](#) doivent considérer un PDPTW incluant des contraintes de précédence entre les requêtes liées à des incompatibilités entre espèces (par exemple à cause de risques de transmission de maladies). Des instances jusqu'à 200 requêtes sont résolues par leur méthode de génération de colonnes.

Une autre méthode, déjà révélée efficace pour le TSP et le TSP asymétrique, apporte de bons résultats sur ce type de problèmes. Il s'agit d'une SEPGCOU appliquée à un modèle de flux où les variables sont les arcs reliant chaque paire de service. La différence par rapport aux méthodes précédemment présentées est que la relaxation linéaire du modèle est alors résolue directement, par un algorithme du type simplexe ou point intérieur. Ceci est possible car le modèle de flux est compact, c'est-à-dire avec un nombre réduit de variables – le nombre d'arcs du graphe en l'occurrence. La difficulté principale de ces méthodes vient du fait que de nombreuses coupes sont nécessaires pour améliorer la borne obtenue par la résolution de la relaxation linéaire, qui est au départ de piètre qualité. [Ruland et Rodin \(1997\)](#) trouvent quatre familles d'inégalités non seulement valides pour le 1-PDP, leur sujet d'étude, mais aussi pour le PDPTW. D'autres inégalités proviennent de travaux sur le *Problème du Voyageur de Commerce Asymétrique avec Contraintes de Précédence* ou *Precedence-Constrained Asymmetric Traveling Salesman Problem* ([Balas et al., 1995](#); [Ascheuer et al., 2000](#)), où la visite d'un nœud peut être conditionnée par la visite précédente d'un ensemble de nœuds. La SEPGCOU de [Lu et Dessouky \(2004\)](#) résout des instances de PDP et PDPTW à 25 requêtes, tandis que celle de Cordeau, avec un modèle de base sensiblement différent permet la résolution d'instances de DARP (PDPTW enrichi de limites sur le temps de transport individuel) à 32 requêtes. Plus récemment, [Cordeau \(2006\)](#) et [Ropke et al. \(2007\)](#) proposent deux nouveaux modèles avec un nombre exponentiel de contraintes et résolvent des problèmes de PDPTW et de DARP à 100 requêtes.

2.1.2.2.3 Heuristiques Comme nous l'avons précédemment écrit, les TAD se prêtent généralement bien à des algorithmes heuristiques. À l'heure actuelle, trois approches se partagent les faveurs des chercheurs. Une approche très usitée est basée sur des algorithmes d'insertion (pouvant être intégrés dans des recherches locales ou tabous). Les algorithmes de construction par insertions sont généralement moins performants en terme d'objectif que les algorithmes de recherche locale. En revanche, ils sont plus robustes (adaptables) et sont naturellement bien conçus pour les problèmes dynamiques. Le deuxième type d'approches proposées par certains auteurs est d'utiliser directement une métaheuristique de type recherche à grand voisinage (LNS). Les méthodes donnant certainement les meilleurs résultats sur des instances de la littérature sont celles en deux phases, la première de *clustering* souvent réalisée par un algorithme génétique ou un recuit simulé, accompagnée d'une phase de *routing* (recherche locale, tabou...).

Sexton et Bodin (1985b) furent parmi les premiers à proposer une méthode de décomposition pour résoudre un DARP à plusieurs véhicules. Ils résolvent le problème de calcul d'itinéraire par une décomposition de Benders où le problème esclave est celui d'horodatage, traité par un algorithme itératif *ad hoc* (Sexton et Bodin, 1985a). La fonction objectif tient compte du temps d'avance à l'arrivée par rapport à une date limite imposée et de la durée de transport. Ce travail est développé avec une fonction objectif plus générale autorisant des retards par Sexton et Choi (1986). Cette idée de décomposition est reprise et adaptée à des problèmes à plusieurs milliers de requêtes, notamment par Dumas et al. (1989), avec la notion de *mini-cluster*. Ioachim et al. (1993) approfondissent cette voie en optimisant la sélection des *clusters*. D'autres chercheurs développent des heuristiques sur un schéma similaire. Xu et al. (2003) traitent un PDP original incluant des contraintes de fenêtres de temps multiples, des contraintes de compatibilité et de temps de conduite maximum pour les chauffeurs.

Une voie longtemps privilégiée par les chercheurs fut l'étude de l'insertion de requêtes dans une tournée. Cette approche permet d'obtenir directement une méthode de construction de tournées très rapide. Ce problème peut intervenir dans d'autres méthodes basées sur l'exploration de voisinages *constructifs/destructifs*. De plus, l'insertion d'une requête dans une tournée est l'enjeu majeur lié à la gestion d'une application dynamique. C'est donc naturellement que les précurseurs en la matière Wilson et al. (1971); Wilson et Weissberg (1976); Wilson et Colvin (1976), calculent les plans de route de différents systèmes de transport aux États-Unis, par des méthodes constructives basées sur l'insertion séquentielle des requêtes. C'est aussi l'option choisie par les concepteurs de systèmes globaux gérant intégralement tous les types de transport de personnes (Horn, 2002; Dial, 1995). De tels systèmes n'existent toujours pas, et les TAD sont plutôt ciblés sur certaines catégories de personnes comme les handicapés. C'est pour un système leur étant destiné, que Roy (1984) présente un algorithme basé sur le même modèle que précédemment. Des tournées sont construites à partir des requêtes connues à l'avance sur des critères de proximité spatiale et temporelle. De nouvelles requêtes peuvent alors être insérées dans la solution courante.

Jaw et al. (1986) proposent un algorithme d'insertion pour un problème intégrant des contraintes de qualité de service. Les passagers spécifient une date de ramassage au plus tôt ou de livraison au plus tard, à laquelle s'ajoute une minimisation de la du-

rée de transport individuelle et des temps d'attente ; sachant que les véhicules ne sont pas autorisés à attendre avec des passagers à l'intérieur. Cette définition du DARP, à quelques nuances près, est assez répandue. L'ordre d'insertion des requêtes est croissant en fonction des dates de ramassage au plus tôt. Les particularités des autres algorithmes d'insertion reposent soit sur une spécificité du problème, soit sur l'ordre de traitement des requêtes, soit sur une procédure d'amélioration de la solution obtenue après l'insertion de toutes les requêtes. Ces algorithmes ont une complexité générale quadratique en fonction du nombre de requêtes.

Toth et Vigo (1996) considèrent un problème de transport pour handicapés à Bologne (300 requêtes par jour). L'objectif est la minimisation du coût du service global, en respectant des fenêtres de temps imposées par les usagers, au ramassage ou à la livraison, ainsi qu'un temps maximal de transport. Différents types de véhicules sont disponibles suivant les passagers. Une procédure d'insertion parallèle assigne les requêtes aux tournées puis effectue des mouvements inter et intra tournées. Une post-optimisation par une recherche tabou granulaire (*Tabu Thresholding* ou *Granular Tabu Search*) est ajoutée dans Toth et Vigo (1997).

Dessouky et al. (2003) ajoutent un critère d'optimisation écologique à ceux usuels. Diana et Dessouky (2004) utilisent, eux, une méthode de *regret insertion*, qui bien qu'élevant la complexité de l'algorithme d'insertion au cube du nombre de requêtes, permet de résoudre des instances à 1000 requêtes. Lu et Dessouky (2006) considèrent un coût d'insertion original. Ils évaluent une insertion en mesurant les détours effectués, mais aussi la perte de liberté sur les fenêtres de temps et les croisements entre les tournées. Ce dernier critère est sensé rendre la solution meilleure aux yeux de l'opérateur par son allure visuelle.

Madsen et al. (1995) adaptent l'algorithme de Jaw et al. (1986) au cas dynamique. Coslovich et al. (2006) construisent des voisinages des tournées en perturbant les données, afin d'augmenter le taux d'acceptation des requêtes à venir. Une modélisation en logique floue permet de traduire l'indéterminisme des temps de transports ou les contraintes horaires d'une application réelle. C'est via un tel modèle que Teodorovic et Radivojevic (2000) évaluent l'insertion des requêtes. Dans le même ordre d'idées, Fu (2002) s'intéresse à des variations stochastiques des durées de transport.

En ce qui concerne la recherche locale, nous citons d'abord quelques travaux sur le 1-PDPTW. Psaraftis (1983b) développe un procédé de mouvements standard (*k-interchange*) dans un cadre non contraint, repris plus tard par Savelsbergh (1990) pour des problèmes contraints. Healy et Moll (1995) dirigent, eux, leur recherche locale vers des solutions à « grand » voisinage plutôt que vers des solutions à « bon » voisinage. Une approche particulière (Van der Bruggen et al., 1993) construit des tournées en ordonnant les requêtes en fonction de leurs fenêtres de temps, ce qui amène éventuellement à des tournées irréalisables. L'objectif de la recherche locale est alors de rendre ces solutions réalisables. Plus récemment, Wolfler Calvo et Colorni (2007) construisent itérativement, après avoir déterminé un nombre de véhicules, des tournées potentiellement irréalisables en résolvant un problème d'affectation ; ils modifient ensuite ces tournées afin de les rendre réalisables. Leur heuristique est rapide sur des instances de 10 à 180

requêtes, générées sur le réseau routier de la ville de Milan. [Xiang et al. \(2006\)](#) considèrent des contraintes supplémentaires pour les conducteurs : pauses, degrés de qualification suivant les requêtes. . . Ils comparent leurs résultats avec les valeurs optimales obtenues par une adaptation de l’algorithme de SEPGCOL de [Dumas et al. \(1989\)](#). Cette heuristique sert de base à la résolution d’un problème dynamique sous certaines contraintes dépendant du temps ou stochastiques ([Xiang et al., 2007](#)).

Dans le cadre du PDPTW, [Bent et Van Hentenryck \(2006\)](#) évitent le problème des optima locaux grâce à une recherche à grand voisinage. Le nombre de routes est tout d’abord minimisé par un recuit simulé. [Ropke et Pisinger \(2006\)](#) ont une approche similaire, leur LNS est intégrée dans un recuit simulé qui contrôle le filtrage des solutions.

D’autres metaheuristiques ont été appliquées au DARP. [Li et Lim \(2001\)](#); [Colorni et al. \(1996\)](#); [Baugh et al. \(1998\)](#) proposent des méthodes basées sur un recuit simulé. [Baugh et al. \(1998\)](#) définissent une fonction combinant dix objectifs dans une somme pondérée. Un algorithme génétique permet d’effectuer la phase de *clustering* dans [Rekiek et al. \(2006\)](#), suivi d’une phase d’insertion, et dans [Jørgensen et al. \(2006\)](#) complété par une recherche locale. Un algorithme génétique est aussi utilisé seul, dans le cas du PDPTW par [Jung et Haghani \(2000\)](#), par [Pankratz \(2003\)](#) pour un DARP et par [Chevrier et al. \(2006b\)](#). Les gènes de ce dernier ont la particularité de représenter un groupe de requêtes au lieu d’une seule requête.

Une metaheuristique réputée efficace en transport et plus particulièrement sur le VRPTW ([Gendreau et al., 1998](#)), est la recherche tabou, qui bien qu’utilisée souvent en complément d’autres méthodes obtient en général de bons résultats seule. [Nanry et Barnes \(2000\)](#); [Lau et Liang \(2002\)](#) proposent des premiers algorithmes pour le PDPTW. [Cordeau et Laporte \(2003b\)](#) développent une recherche tabou prenant en compte des contraintes de temps maximal de transport et de durée maximale de tournée. Pour gérer ces contraintes, ils autorisent en cours de résolution l’apparition de tournées irréalisables. Enfin, [Melachrinoudis et al. \(2007\)](#) considèrent des fenêtres de temps souples et minimisent l’inconvénient occasionné par la déviation par rapport à des dates de service souhaitées.

2.2 Méthode de génération de colonnes

Dans cette section, nous présentons les principes de la méthode de résolution par génération de colonnes que nous utilisons pour résoudre le DARP. Cette méthode de décomposition est souvent utilisée en transport sous forme de méthode exacte mais aussi en tant qu’heuristique. Pour une vision détaillée de la génération de colonnes et de ses applications en transport nous conseillons au lecteur les articles de [Lübbecke et Desrosiers \(2005\)](#), [Barnhart et al. \(1998\)](#) et le livre de [Desaulniers et al. \(2005\)](#).

2.2.1 Principes de la méthode

La méthode de génération de colonnes permet de résoudre des programmes linéaires. Les variables sont appelées colonnes. Lorsque la génération de colonnes est intégrée dans une recherche arborescente, lors de la résolution de programmes linéaires en nombres entiers par exemple, on parle de SEPGCOL (*Branch & Price* en anglais). C'est une méthode itérative. À chaque itération, on résout un problème restreint (en nombre de colonnes). De la solution obtenue et des principes de dualité de la programmation linéaire, on déduit l'optimalité ou non de la solution obtenue. À l'itération suivante on enrichit le problème restreint d'une ou plusieurs colonnes. Les informations obtenues lors de la résolution précédente permettent aussi de caractériser des variables intéressantes à ajouter. Le problème initial à résoudre est appelé problème maître ou principal et le générateur de nouvelles colonnes est appelé problème esclave ou sous-problème.

Considérons le problème maître générique MP, où les données a_ω^r et b^r sont des rationnels et c_ω des réels. Les variables de décision sont les réels λ_ω .

$$(MP) \quad \min \sum_{\omega \in \Omega} c_\omega \lambda_\omega \quad (2.1)$$

s.c.q.

$$\sum_{\omega \in \Omega} a_\omega^r \lambda_\omega \leq b^r \quad \forall r = 1, \dots, R, \quad (2.2)$$

$$\lambda_\omega \geq 0 \quad \forall \omega \in \Omega. \quad (2.3)$$

Plaçons nous à l'itération it de la résolution par génération de colonnes. On résout optimalement le problème maître restreint à $\Omega^{it} \subseteq \Omega$. On suppose que la méthode de résolution utilisée donne aussi une solution optimale au dual du programme linéaire³. Notons π^r la variable duale associée à la $r^{\text{ième}}$ contrainte (2.2). Toute colonne $\omega \in \Omega \setminus \Omega^{it}$ susceptible de diminuer la valeur de la fonction objectif si on l'ajoute au problème maître restreint a un coût réduit c_ω^π (défini par l'équation 2.4) négatif. Le problème esclave consiste donc à trouver de telles colonnes. Il est souvent posé comme un problème de minimisation sur les c_ω^π . Si aucune colonne de coût réduit négatif n'existe, on déduit alors l'optimalité pour MP de la solution obtenue à l'itération it .

$$c_\omega^\pi = c_\omega - \sum_{r=1}^R a_\omega^r \pi^r \quad (2.4)$$

Le schéma de la figure 2.1 illustre le déroulement de l'algorithme de génération de colonnes.

Le fait d'augmenter progressivement la taille du problème à résoudre rend la génération de colonnes particulièrement adaptée pour la résolution de problèmes possédant

³Les algorithmes, basés sur celui du simplexe ou celui du point intérieur, implémentés dans la plupart des solveurs du marché donnent les deux solutions.

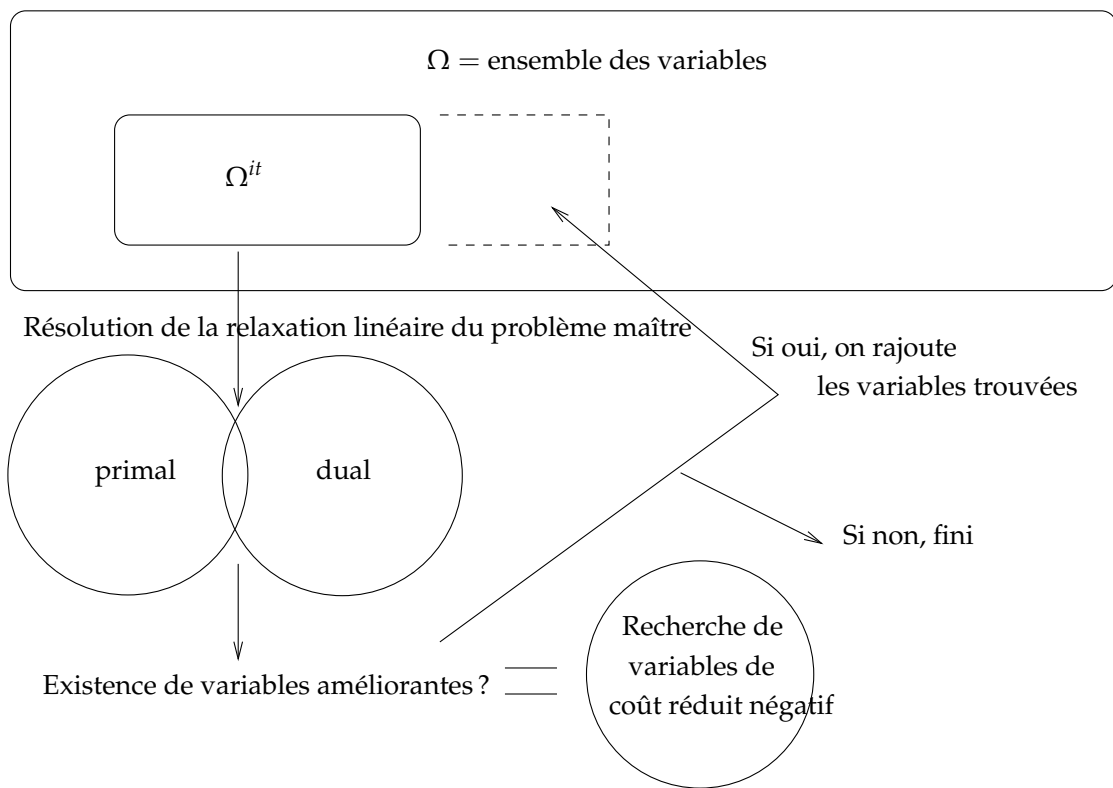


FIG. 2.1: Méthode de génération de colonnes.

un très grand nombre de colonnes. On peut ainsi espérer obtenir une solution optimale en ayant considéré qu'un sous-ensemble des colonnes.

2.2.2 Contexte d'utilisation – Calcul de bornes inférieures pour des programmes linéaires en nombres entiers

La génération de colonnes est naturellement adaptée aux problèmes ayant un grand nombre de variables. Cependant, dans le cadre de la résolution de programme linéaires en nombre entiers, même lorsque le problème se modélise avec un nombre raisonnable de variables (modèle *compact*) – permettant une résolution directe – il peut être intéressant d'utiliser un modèle équivalent, dit *étendu*, possédant un grand nombre de variables. En effet, ce dernier peut fournir une meilleure relaxation linéaire.

Considérons le programme linéaire en nombres entiers *compact* IP et P sa relaxation linéaire, possédant deux types de contraintes A , que nous qualifions de globales, et D , que nous qualifions de locales. Nous supposons x et c des vecteurs de réels de taille n et A, D, b et d à valeurs rationnelles.

$$(IP) \quad \min c^T x \quad (2.5)$$

s.c.q.

$$Ax \geq b, \quad (2.6)$$

$$Dx \geq d, \quad (2.7)$$

$$x \in \mathbb{N}^n. \quad (2.8)$$

L'optimisation de P se fait sur le polyèdre

$$Q_P = \{x \in \mathbb{R}_+^n : Ax \geq b\} \cap \{x \in \mathbb{R}_+^n : Dx \geq d\}.$$

Le passage de P à un programme linéaire possédant un grand nombre de variables (MP) se fait par une transformation basée sur un changement de variables appelé « décomposition de Dantzig-Wolfe » qui introduit les colonnes λ_ω comme des objets complexes composés d'éléments de $\{x \in \mathbb{N}^n : Dx \geq d\}$. Les colonnes de MP restent soumises aux contraintes A . L'optimisation de MP se fait donc sur le polyèdre

$$Q_{MP} = \{x \in \mathbb{R}_+^n : Ax \geq b\} \cap \text{conv}(\{x \in \mathbb{N}^n : Dx \geq d\}),$$

où $\text{conv}(E)$ désigne l'enveloppe convexe de E , qui est inclus dans Q_P car on a

$$\text{conv}(\{x \in \mathbb{N}^n : Dx \geq d\}) \subset \{x \in \mathbb{R}_+^n : Dx \geq d\}.$$

On dit qu'un polyèdre (sur \mathbb{R}_+^n) possède la *propriété d'intégralité* s'il est équivalent à l'enveloppe convexe de sa restriction sur \mathbb{N}^n . On en déduit que MP fournit une meilleure borne inférieure à IP que P si les contraintes locales forment un polyèdre ne vérifiant pas la *propriété d'intégralité*.

Un autre intérêt lié à l'utilisation d'un modèle *étendu* est qu'il permet de « casser » la symétrie parfois apparente dans les contraintes locales. Par exemple, lorsque la matrice D a une structure en blocs diagonaux.

Il est notable que la relaxation lagrangienne sur les contraintes locales des éléments de Ω donne la même valeur de relaxation linéaire que le modèle étendu.

2.2.2.1 Intégration dans une recherche arborescente

Une méthode de résolution efficace pour les programmes linéaires en nombre entiers est basée sur la combinaison de la résolution de la relaxation linéaire du problème et de la méthode classique de séparation et évaluation progressive. À chaque nœud de l'arbre de recherche on résout un programme linéaire donnant dans le cas d'une minimisation, une borne inférieure au problème. Lorsque cette résolution se fait par génération de colonnes, on parle de SEPGCOL. Les méthodes de branchement classiques doivent alors être adaptées à ce modèle.

Considérons le problème IMP correspondant à MP enrichi de contraintes d'intégralités sur les variables λ_ω . Brancher directement sur la valeur des variables λ_ω est souvent une mauvaise stratégie car de telles règles sont soit faibles soit génératrices d'arbres déséquilibrés, à cause du très grand nombre de variables. De plus, de telles règles sont souvent difficiles à intégrer dans le problème esclave. On préfère généralement brancher sur d'autres variables permettant de caractériser les éléments de Ω , par exemple en utilisant le modèle compact.

2.2.3 Paramétrages – Efficacité

2.2.3.1 Convergence

Un des problèmes rencontré par la génération de colonnes est la variance erratique des valeurs des variables duales en début de résolution, qui sont aussi peu contraintes que le nombre de colonnes est petit. Ainsi de mauvaises colonnes sont générées sur des paramètres quasiment aléatoires avec un impact négatif sur la convergence vers les colonnes optimales. Une approche proposée pour parer ce défaut est de « stabiliser » les valeurs prises par les variables duales, d'où le nom de méthodes de stabilisation. Pour cela, on pénalise dans la fonction objectif, l'écart entre les valeurs courantes des variables duales par rapport à des valeurs fixées à chaque itération. Pour de plus amples informations, nous dirigeons le lecteur vers les articles [Ben Amor et al. \(2006\)](#) et [Ben Amor et Desrosiers \(2006\)](#). Une autre approche équivalente est de considérer une solution duale qui soit une combinaison de solutions optimales ([Rousseau et al., 2007](#)).

Lors de la résolution de programmes linéaires en nombre entiers par génération de colonnes, il arrive souvent que la résolution optimale de la relaxation linéaire du problème maître restreint tarde à converger, tout en restant proche de l'optimum – ce dernier pouvant même être atteint. Une solution simple pour éviter cet inconvénient est

de brancher prématurément plutôt que d'attendre l'obtention d'une solution optimale, et de sa preuve d'optimalité, pour la relaxation linéaire du problème maître restreint.

2.2.3.2 Gestion des colonnes

Le problème maître restreint peut par sa taille ou par symétrie devenir difficile à résoudre. Supprimer des colonnes a priori inutiles est toujours possible, puisque si des colonnes supprimées s'avèrent utiles à une itération ultérieure, elles seront alors naturellement régénérées. Le nombre de colonnes remontées – intégrées au problème maître restreint – à chaque itération du générateur de colonnes est un paramètre influent sur la rapidité de l'algorithme.

2.2.3.3 Initialisation

Pour amorcer la méthode, il est toujours possible de créer une colonne fictive – permettant de satisfaire toutes les contraintes du problème maître – qu'on pénalise fortement dans la fonction objectif, lorsqu'on dispose d'un ensemble Ω^0 vide. Un bon ensemble de colonnes de départ peu évidemment rendre la résolution plus facile. Cependant, même initialisée par une solution optimale, la preuve d'optimalité – liée à la construction de la base d'une solution optimale – peut s'avérer difficile à obtenir. Il peut être plus profitable d'utiliser un ensemble Ω^0 de colonnes très diversifiées.

2.2.3.4 Résolution du problème esclave

La difficulté du problème esclave est parfois cruciale pour la génération de colonnes. Il peut être intéressant de résoudre ce problème de façon heuristique, surtout lorsque l'ensemble de colonnes à coût réduit négatif est élevé. Il est alors envisageable d'utiliser les colonnes déjà générées pour en construire de nouvelles.

Pour de nombreux problèmes de calcul de tournées de véhicules, le problème esclave prend la forme d'un Problème de Plus Court Chemin avec Contraintes de Ressources.

2.3 Plus court chemin avec contraintes de ressources (SPPRC)

Les problèmes de plus court chemin entre deux sommets dans des graphes pondérés sont très bien connus depuis longtemps et des algorithmes polynomiaux sont disponibles pour différents types de graphes : Dijkstra dans le cas de poids positifs, Bellman dans le cas général... Le fait de consommer en cours de chemin des ressources disponibles en quantités limitées (SPPRC) rend le problème NP-difficile. Et l'ajout d'une contrainte d'élémentarité sur le chemin (ESPPRC) rend le problème NP-difficile au sens

fort, comme réduction du problème de *Sequencing Within Intervals* (Dror, 1994). La figure 2.2 représente un SPPRC entre v_0 et v_3 avec une seule ressource dont la consommation cumulée est contrainte en chaque sommet (la valeur à ne pas dépasser est indiquée prêt de chaque sommet).

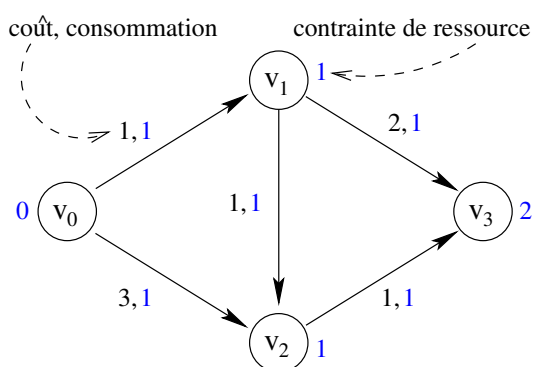


FIG. 2.2: Un SPPRC à une ressource

Le SPPRC et le ESPPRC nous intéressent particulièrement car ils interviennent à plusieurs reprises dans cette thèse. Le cas principal où ces problèmes apparaissent est lors de la résolution du problème esclave du schéma de génération de colonnes adapté au DARP. Cette approche, dont le point clef est la résolution du problème esclave, est détaillée dans le chapitre 4.1. Dans cette section, nous restreignons notre discours au cas de ressources dont la quantité consommée se cumule au fur et à mesure du chemin. Ce type de ressources est assez représentatif des ressources dont la fonction d'extension est croissante. La fonction d'extension est la fonction permettant de calculer la consommation d'une ressource pour un chemin après l'avoir étendu d'un arc. Pour plus de détails nous guidons le lecteur vers Irnich et Desaulniers (2005) et Desrosiers et al. (1995). Nous concentrons aussi notre étude sur la résolution exacte du problème qui est toujours indispensable pour conclure la génération de colonnes à l'optimalité. L'utilisation d'heuristiques sur le problème esclave est une voie explorée avec succès. Néanmoins, ces dernières sont souvent basées sur des relaxations résolues par un algorithme basé sur une méthode exacte. Les approches purement heuristiques sont plus rares, nous pouvons citer les travaux de Savelsbergh et Sol (1998) et de Xu et al. (2003) qui effectuent une recherche locale – en se basant sur les colonnes de base de la résolution du problème maître restreint qui ont la propriété d'avoir un coût réduit nul. La littérature propose différentes approches exactes pour les problèmes de plus court chemin avec contraintes de ressources dont des méthodes basées sur de la programmation dynamique (Desrochers, 1988; Righini et Salani, 2005) ainsi que de la programmation par contraintes (Rousseau et al., 2004). Un cas particulier, appelé problème de plus court chemin contraint (CSPP), où les contraintes sur les ressources sont globales à tous les sommets, est résolu par relaxation lagrangienne par plusieurs auteurs dont Beasley et Christofides (1989) et Borndörfer et al. (2001). Nous présentons ici un algorithme basé sur celui proposé par Desrochers (1988) pour le cas du problème de plus court chemin avec fenêtres de temps.

Après avoir défini formellement le problème, nous présentons un algorithme standard de programmation dynamique qui sert de base pour plusieurs algorithmes développés plus loin dans la thèse.

2.3.1 Modélisation d'un ESPPRC

Soit $G = (\mathcal{V}, \mathcal{A}, \mathcal{W})$ un graphe multivalué où on note les sommets v_i ($i = 0, \dots, V$). Le sommet v_0 est la source et le sommet v_V le puits. Chaque arc (v_i, v_j) a une consommation $c_{ij}(w)$ sur la ressource w ($w = 0, \dots, W$). La ressource 0 correspond par définition au coût. Une contrainte sur chaque ressource, hormis la ressource 0, est associée à chaque sommet. La consommation cumulée le long d'un chemin jusqu'au sommet v_i est notée $c_i(w)$. La contrainte de consommation de la ressource w sur le sommet v_i prend la forme de l'intervalle $[A_w^i, B_w^i]$. En général, A_w^i est un seuil de consommation ; c'est-à-dire qu'au sommet i on réajuste $c_i(w)$ si nécessaire par l'équation (2.9).

$$c_i(w) := \max(c_i(w), A_w^i) \quad (2.9)$$

L'objectif (équation (2.10)) est de trouver un chemin de coût minimal de v_0 vers v_V parmi les chemins réalisables sur l'ensemble des ressources. La variable de décision δ_{ij} est fixée à 1 si l'arc (v_i, v_j) appartient à une solution et 0 sinon.

(EP)

$$\min \sum_{(v_i, v_j) \in \mathcal{A}} c_{ij}(0) \delta_{ij} \quad (2.10)$$

s.c.q.

$$\sum_{(v_i, v_j) \in \mathcal{A}} \delta_{ij} - \sum_{(v_j, v_i) \in \mathcal{A}} \delta_{ji} = 0 \quad \forall j = 1, \dots, V-1, \quad (2.11)$$

$$\sum_{(v_i, v_j) \in \mathcal{A}} \delta_{ij} \leq 1 \quad \forall i = 0, \dots, V, \quad (2.12)$$

$$\sum_{(v_0, v_j) \in \mathcal{A}} \delta_{0j} = 1, \quad (2.13)$$

$$A_w^i \leq c_i(w) \leq B_w^i \quad \forall i = 0, \dots, V; \forall w = 1, \dots, W, \quad (2.14)$$

$$\delta_{ij}(c_i(w) + c_{ij}(w) - c_j(w)) \leq 0 \quad \forall (v_i, v_j) \in \mathcal{A}; \forall w = 1, \dots, W, \quad (2.15)$$

$$\delta_{ij} \in \{0, 1\} \quad \forall (v_i, v_j) \in \mathcal{A}. \quad (2.16)$$

Les inégalités (2.11) représentent les contraintes de flot assurant la continuité du chemin. L'élémentarité du chemin est assurée par les contraintes (2.12). L'existence d'un chemin unique issu de la source est imposée par la contrainte (2.13). Les contraintes (2.14) contrôlent la consommation cumulée sur chaque ressource et les contraintes non linéaires (2.15) la continuité du flot de consommation des ressources. On admet que toutes les ressources se cumulent par incrémentation du poids de l'arc associé à la ressource dans la limite du seuil autorisé.

2.3.2 Résolution par programmation dynamique

L'algorithme 1 part de la source et construit pas-à-pas des chemins vers la destination. La construction se fait par extension de chemins partiels. Un chemin partiel a pour origine la source et pour extrémité terminale un sommet quelconque du graphe. L'extension d'un chemin partiel cp se fait par le calcul des chemins partiels correspondant à cp augmenté par un arc sortant de l'extrémité terminale de cp . Les nouveaux chemins partiels étendus sont rejetés s'ils violent au moins une contrainte. Sinon chaque chemin partiel est comparé avec ceux ayant la même extrémité terminale ; les chemins partiels dominés sont éliminés. Il y a dominance entre deux chemins partiels si un des deux domine l'autre pour chaque ressource prise individuellement. On admet qu'après extension vers le sommet i , le niveau de consommation d'une ressource ($c_i(w)$) est ajusté à sa borne inférieure (A_w^i) s'il y est inférieur. On note $ref_w(ch, (i, j))$ la valeur de la fonction d'extension de la ressource w du chemin partiel ch par l'arc (v_i, v_j) . La formule (2.17) donne les règles d'extension des ressources ; la consommation d'un chemin partiel ch sur une ressource w est notée $c_{ch}(w)$.

$$ref_w(ch, (v_i, v_j)) = \max \left\{ c_{ch}(w) + c_{ij}(w), A_w^i \right\} \quad (2.17)$$

La complexité de cet algorithme dépend donc des ressources considérées. Si la consommation des ressources prend des valeurs entières et que le nombre de valeurs prises est borné par un nombre indépendant de la taille des données, cet algorithme est pseudo-polynomial.

La figure 2.3 illustre le déroulement de l'algorithme sur un graphe à quatre sommets et deux ressources dont le coût. Un chemin partiel est identifié par son nom et son niveau de consommation des ressources entre accolades, sur le modèle

$$ch_i \{ \text{coût}, \text{autre ressource} \}.$$

L'objectif est de minimiser le coût des chemins partant de v_0 en respectant les contraintes de consommation de ressource en chaque sommet, indiquées entre crochets sur la figure 2.3. À chaque itération, on étend le chemin partiel marqué par une étoile dans \mathcal{CH} . Le chemin partiel ch_3 issu de ch_1 est supprimé car il consomme 2 unités sur la ressource qui est limitée à 1 au sommet v_2 . Le chemin partiel ch_5 est quant à lui supprimé car il est dominé par ch_4 .

Les auteurs utilisant des variantes de cet algorithme pour résoudre leur problème esclave, apportent des spécifications sur les contraintes considérées ou relâchées et sur les critères de dominance.

Desrochers (1988) est sans doute le premier à l'utiliser en transport pour calculer des itinéraires de bus scolaires. Desrosiers et al. (1986) et Savelsbergh et Sol (1998) reprennent cet algorithme pour un PDPTW et Dumas et al. (1989) pour un DARP. Righini et Salani (2006, 2005) proposent d'utiliser de façon symétrique l'algorithme, en partant de la source et de la destination, afin de réduire le nombre de chemins partiels à construire. Enfin, Ropke (2005) l'utilise dans différentes versions heuristiques

Algorithme 1 : Algorithme de programmation dynamique

Données : G : un graphe multivalué sur $W + 1$ ressources ; \mathcal{CH} : une liste de chemins partiels non traités ; CH_i : la liste des chemins partiels d'extrémité terminale i

Résultat : CH_V

```

1 Initialisation :  $\mathcal{CH} \leftarrow$  chemin partiel réduit à  $\{0\}$  ;
2 tant que  $\mathcal{CH} \neq \emptyset$  faire
3   prendre  $ch \in \mathcal{CH}$  ;
4    $et_{ch} :=$  extrémité de  $ch$  ;
5    $domine := faux$  ;
6   pour chaque chemin partiel  $ch' \in CH_{et_{ch}}$  faire
7     si  $ch(w) \leq ch'(w), \forall w = 0, \dots, W$  alors
8        $CH_{et_{ch}} := CH_{et_{ch}} \setminus \{ch'\}$  ;
9     sinon
10      si  $ch'(w) \leq ch(w), \forall w = 0, \dots, W$  alors
11         $domine := vrai$  ;
12      stop ;
13 si  $domine = faux$  alors
14    $CH_{et_{ch}} := CH_{et_{ch}} \cup \{ch\}$  ;
15   pour chaque arc  $(et_{ch}, i)$  faire
16      $ch' :=$  l'extension de  $ch$  par  $(v_{et_{ch}}, v_i)$  ;
17     si  $ch'$  est valide alors
18        $\mathcal{CH} := \mathcal{CH} \cup \{ch'\}$  ;

```

2.3. Plus court chemin avec contraintes de ressources (SPPRC)

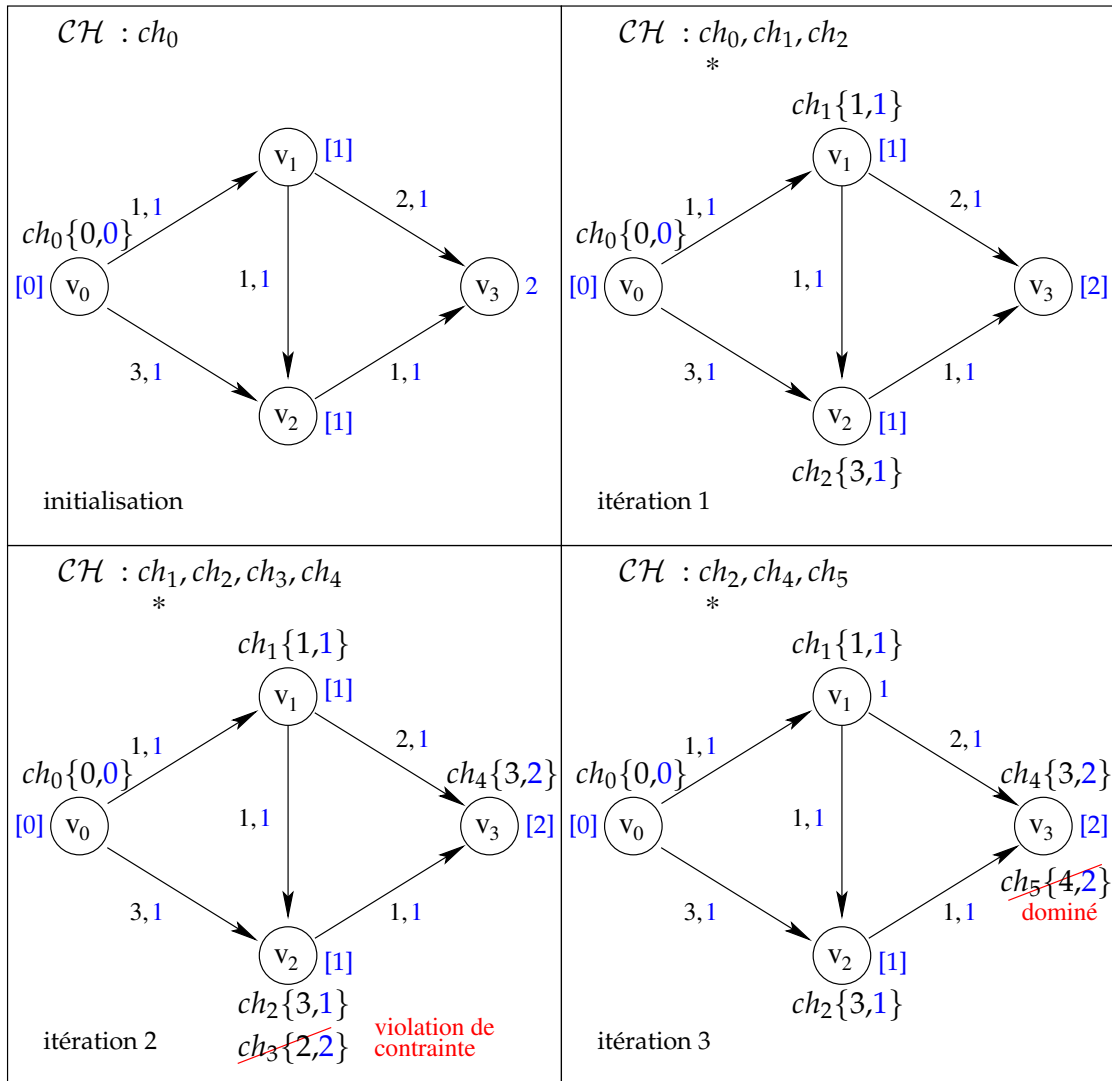


FIG. 2.3: Trace de l'algorithme de programmation dynamique

pour des problèmes de PDPTW et de DARP. [Feillet et al. \(2004\)](#) et [Chabrier \(2006\)](#) proposent aussi une version adaptée au cas de recherche de chemins élémentaires alors que d'autres auteurs ([Irnich et Desaulniers, 2005](#)) interdisent les cycles excédant une longueur maximale.

Des critères de qualité de service autres que les fenêtres de temps peuvent se représenter sous forme de ressources additionnelles dans le cadre d'un ESPPRC.

Deuxième partie

Qualité de service pour le transport à la demande

Chapitre 3

Critères de qualité de service

RÉSUMÉ : *Après avoir défini formellement le problème étudié dans la section 3.1, nous proposons dans la section 3.2 une liste de critères de qualité de service, organisée suivant le niveau à partir duquel ils sont mesurables : à partir d'une longue période de fonctionnement, d'une solution complète, d'une tournée, de la course d'un passager ou d'un simple tronçon de route. Nous donnons une modélisation formelle de ces critères lorsque cela nous semble pouvoir rentrer dans le cadre de notre étude plus concentrée sur les critères mesurables à une petite échelle.*

Nous considérons dans la section 3.3 l'optimisation des critères mesurables au moins sur une tournée, dans le cas où un véhicule connaît exactement les lieux qu'il doit visiter ainsi que l'ordre dans lequel ils doivent être visités. Le problème se restreint alors à un problème d'horodatage de la tournée, et/ou de choix des itinéraires à emprunter entre chaque paire de lieux consécutifs.

3.1 Définition du problème étudié

Nous définissons dans ce chapitre un cas épuré de DARP – en fait un cas particulier de PDPTW – qui sert de base à l'étude de l'optimisation de critères de qualité de services dont nous effectuons la présentation dans la section 3.2. Ici, les notations sont introduites mais le DARP n'est pas présenté sous forme d'un modèle d'optimisation.

Comme nous l'avons précédemment écrit, il n'y a pas de définition formelle rigide du DARP. Si on reste dans l'idée qu'un DARP est un PDPTW classique (considérant un ensemble de requêtes de services, chacune étant définie par un ramassage et une livraison) et enrichi de quelques contraintes ou objectifs, on peut proposer le DARP générique suivant. Dans le système considéré, chaque utilisateur émet une requête de transport définie par un arrêt pour le ramassage, un arrêt pour la livraison, une quantité de passagers. Le système doit traiter R requêtes, à l'aide d'une flotte homogène de K véhicules. Toutes les requêtes doivent être servies. On dispose du graphe $G = (\mathcal{V}, \mathcal{A})$ où chaque sommet $v \in \mathcal{V}$ correspond à un service, c'est-à-dire à un ramassage, une

livraison ou l'arrivée (le départ) à un dépôt. v_{+r} et v_{-r} identifient respectivement le ramassage et la livraison de la requête de transport r ($r = 1, \dots, R$). On pose $r^+ = r$ et $r^- = -r$. On note v_{R+k} et $v_{-(R+k)}$ les sommets des services de départ et d'arrivée au dépôt du véhicule k . Ainsi l'ensemble \mathcal{R}^+ est l'ensemble des entiers de 1 à R (ramassages), \mathcal{R}^- celui des entiers de $-R$ à -1 (livraisons) et $\mathcal{R}^0 = [-R - K, -R - 1] \cup [R + 1, R + K]$ (dépôts d'arrivée et de départ). Chaque service i a une durée non nulle s_i .

Chaque arc (v_i, v_j) de \mathcal{A} correspond à un tronçon reliant v_i à v_j . Il est pondéré par sa durée T_{ij} et son flux de passagers $F_{ij} = F_i$, c'est-à-dire le nombre de personnes montant ou l'opposé du nombre de personnes descendant en i (F_i est alors négatif). Un véhicule ne peut être affecté qu'à une seule tournée sur l'horizon de temps considéré $([0, T])$.

Pour toute requête, le ramassage doit être effectué avant la livraison et par le même véhicule.

Pour chaque sommet v_i , la date de début de service t_i , aussi appelée date de service, doit respecter une fenêtre de temps $[T_i^{inf}, T_i^{sup}]$.

L'occupation du véhicule à la fin du service i , notée f_i , est elle aussi soumise à une contrainte de la forme $[F_i^{inf}, F_i^{sup}]$ liée à la capacité F du véhicule et dépendant de la nature du service i , voir l'équation (3.1).

$$[F_i^{inf}, F_i^{sup}] = \begin{cases} [0, 0] & \text{si } i \in \mathcal{R}^0 \\ [F_i, F] & \text{si } i \in \mathcal{R}^+ \\ [0, F + F_i] & \text{si } i \in \mathcal{R}^- \end{cases} \quad (3.1)$$

Une tournée ω est attribuée à un véhicule donné par $k(\omega)$.

On note w_{ij} , dont l'expression est donnée par l'équation (3.2), le temps d'attente entre deux services i et j successifs relativement à une solution.

$$w_{ij} = t_j - T_{ij} - t_i - s_i \quad (3.2)$$

Les critères de qualité de service sont intégrés au niveau du calcul du coût des tournées ou sous forme de contraintes.

Traditionnellement, la qualité de service est mesurée par son inverse, c'est-à-dire en terme d'inconvénient.

Afin de simplifier la description formelle des critères de qualité de service pour une tournée ω fixée contenant les services i et j , nous définissons par les équations (3.3) et (3.4) les ensembles $\hat{\omega}_{ij}$ comme l'ensemble des sommets de ω présents entre v_i et v_j et $\bar{\omega}_{ij}$ comme l'ensemble des arcs entre v_i et v_j . Les variables binaires ρ_ω^x et δ_{xy} indiquent respectivement la présence du service x dans la tournée ω et la sélection de l'arc (v_x, v_y) .

$$\hat{\omega}_{ij} = \{v_x \in \mathcal{V}, \rho_\omega^x = 1 : t_i \leq t_x \leq t_j\} \quad (3.3)$$

$$\bar{\omega}_{ij} = \{(v_x, v_y) \in \mathcal{A} : v_x, v_y \in \hat{\omega}_{ij}, \delta_{xy} = 1\} \quad (3.4)$$

Sur la figure 3.1, est représentée une tournée ω . D'après les définitions, on a sur cet exemple : $\dot{\omega}_{+2,-1} = \{+2; +1; -2; -1\}$ et $\bar{\omega}_{+2,-1} = \{(v_{+2}, v_{+1}); (v_{+1}, v_{-2}); (v_{-2}, v_{-1})\}$.

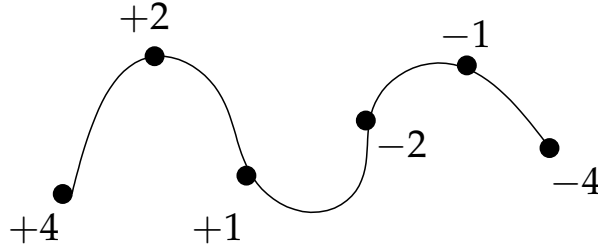


FIG. 3.1: Tournée ω du véhicule $k(\omega) = 2$

Lorsqu'il n'y a aucune ambiguïté sur le véhicule utilisé, nous notons les séquences $\dot{\omega}_{R+k,-R-k}$ et $\bar{\omega}_{R+k,-R-k}$ respectivement $\dot{\omega}$ et $\bar{\omega}$.

L'ensemble des tournées d'une solution est noté Ω^* .

Ce problème simplifié sert de base à la définition des critères de qualité de service.

3.2 Critères de qualité de service

Dans cette section, nous proposons de déterminer précisément un certain nombre d'indicateurs de qualité de service pour un TAD. Nous utilisons le terme de qualité de service dans un sens très large qui englobe les intérêts de tous les acteurs du système : passagers, chauffeurs, transporteurs, collectivités locales...

Que le système de transport à la demande soit exploité par des collectivités locales ou par des entreprises à but lucratif, le calcul des tournées de véhicule doit tenir compte de critères de coûts financiers ; que ce soient des minimisations et des plafonnements de coûts d'exploitation ou des maximisations et des seuils de profits. À cela, on oppose généralement des critères de qualité de service offerts aux passagers, induits par des objectifs sociaux, de confort... Néanmoins, le respect de certains de ces critères a clairement un impact économique positif à plus ou moins long terme sur le TAD. Par exemple, il vaut mieux éviter de rendre des passagers très mécontents car cela peut faire fortement chuter la demande et par suite la rentabilité. Pour ce qui est du coût des tournées calculées, ils restent pénalisant. Cependant, des critères de qualité de service comme le regroupement des passagers dans un véhicule peuvent directement amener une réduction du coût des solutions.

De telles considérations nous amènent à proposer la classification des critères de qualité de service suivant leur niveau d'agrégation. Nous distinguons trois catégories de critères de qualité de service. La première catégorie regroupe les critères qui se mesurent par une fonction de leur évaluation indépendante pour chaque passager, comme par exemple la somme des temps d'attente ou des durées des courses. Parmi

eux, certains, comme le respect des fenêtres de temps, peuvent même s'évaluer indépendamment pour chaque service. La seconde catégorie est celle des critères observables uniquement au niveau d'une tournée ou d'une course comme le temps de pause des chauffeurs. La troisième catégorie concerne les critères mesurables uniquement au niveau de la solution complète, citons par exemple l'équilibrage de charge de travail entre les chauffeurs, la répartition géographique des véhicules et le nombre de requêtes servies. Enfin, une dernière catégorie que nous excluons de notre étude est celle des critères évalués sur un ensemble de solutions comme la régularité – dans le temps – des tournées. Le tableau 3.1 propose une classification des critères de qualité de service que nous évoquons. Chacun de ces critères est détaillé par la suite.

évaluation	critère
tronçon	Écart par rapport à des dates de service souhaitées Temps d'attente absurdes Ordre des services en un même lieu Différents niveaux de qualité de service par type de véhicules Distance parcourue Durée Pénibilité
course	Le déroulement d'une course Pérégrinations Regroupement Régularité des rencontres Sinuosité
tournée	Temps de repos
solution	Taux de remplissage Sensibilité aux perturbations Forme des tournées Nombre de véhicules Équilibrage des charges Différenciation des passagers
long terme	Temps d'obtention de la réponse du système et délai pour l'acceptation d'une nouvelle requête

TAB. 3.1: Classification des critères de qualité de service par niveau d'agrégation

Dans une méthode de décomposition de type problèmes maître/esclave certaines catégories de critères peuvent donc être gérées uniquement dans le problème esclave ; pour peu que leur évaluation combinée soit compatible avec le problème maître.

Un deuxième point important est le mode de prise en compte d'un critère de qualité de service. Dans le problème proposé les contraintes de qualité de service sont soit *dures* et appelées contraintes, soit *souples* présentes dans la fonction objectif. L'optimisation de critères non-linéaires est généralement difficile. Néanmoins, comme nous le montrons dans cette section bon nombre de critères de qualité de service s'expriment de façon linéaire. Pour le cas particulier des critères de qualité de service qui s'obtiennent

par combinaison d'évaluation individuelles deux types d'objectifs sont cruciaux : celui des sommes pondérées et celui des objectifs *maximini*. Le cas *maximini* n'est jamais évoqué car il s'apparente en terme de résolution au cas de contraintes dures.

3.2.1 Évaluation sur un tronçon ou un arrêt

1. Écarts par rapport à des dates de service souhaitées

Les passagers peuvent avoir des préférences ou des impératifs sur les dates de réalisation des services. Le cas général donne ainsi lieu à une fonction $\phi_r(t_{r+}, t_{r-})$ associant un niveau d'inconvénient aux dates de réalisation des services.

Dans certaines applications l'inconvénient se mesure par rapport à un seul service soit le ramassage (notion de *inbound request*, par exemple pour le retour chez soi), soit la livraison (notion de *outbound request*, par exemple pour le départ de chez soi). D'une fonction exprimée par rapport à la date de livraison, on peut déduire un inconvénient lié à la date de ramassage. La réciproque est vraie.

Un cas particulier courant de telles fonctions d'inconvénient est celui des fenêtres de temps, où l'inconvénient est infini à l'extérieur de la fenêtre.

Cette fonction peut prendre des formes diverses (fig. 1) généralement date butoir, fonction convexe, souvent linéaire et limitée dans une fenêtre de temps. Les fonctions d'inconvénient peuvent être intégrées au modèle dans la fonction objectif ou dans des contraintes de majoration. Il est souvent possible de se ramener à une

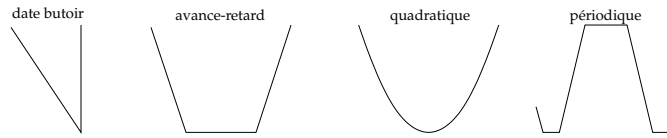


FIG. 3.2: Représentation de la pénalité associée à l'écart par rapport à des horaires idéaux de début de service.

fonction où l'inconvénient est indépendant pour chaque service. Dans ce cas, il existe une fonction ϕ séparable et des fonctions ϕ_{r+} et ϕ_{r-} telle que

$$\forall r \in \mathcal{R}, \phi_r(t_{r+}, t_{r-}) = \phi(\phi_{r+}(t_{r+}), \phi_{r-}(t_{r-})).$$

Par exemple, une pénalisation linéairement liée à la durée d'une course (équation (3.5)) est décomposable en deux fonctions indépendantes. La figure 3.3 représente un tel cas particulier.

$$\forall r \in \mathcal{R}, \phi_r(t_{r+}, t_{r-}) = a(t_{r-} - t_{r+}) + b = [a(t_{r-}) + b] + [-a(t_{r+})] \quad (3.5)$$

2. Temps d'attente absurdes

Pour bon nombre de TAD, aucun passager n'a intérêt à attendre dans le véhicule à l'arrêt avant de descendre même s'il arrive avant la date souhaitée, ou bien d'attendre à l'arrêt après être monté dans le véhicule même s'il monte à la date souhaitée. La figure 3.4 indique les cas où il ne doit pas y avoir d'attente ; c'est-à-dire

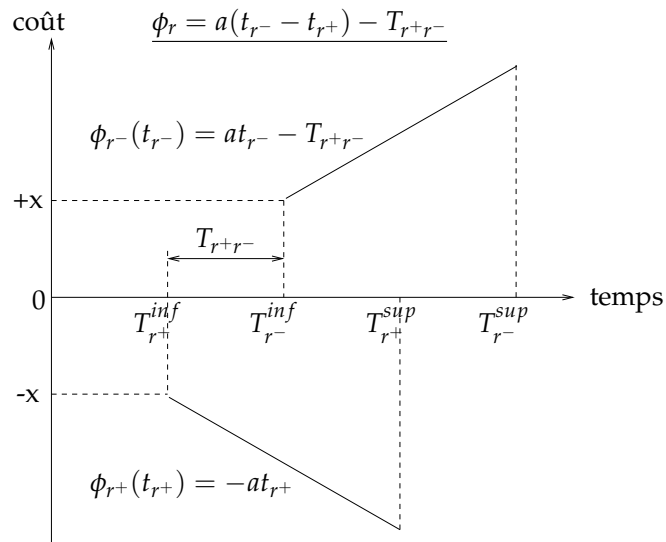


FIG. 3.3: Représentation de la pénalité associée à la durée d'un trajet par des fonctions du temps associées à chaque arrêt

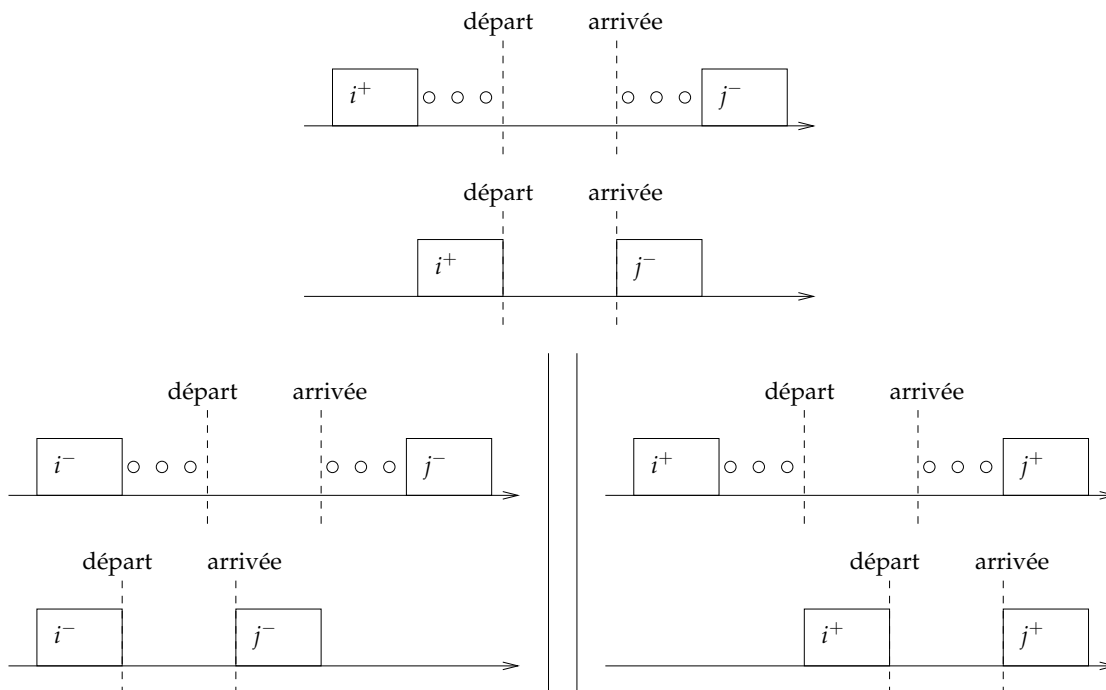


FIG. 3.4: Temps d'attente absurdes

que quelque soit la fonction objectif de l'optimisation, des solutions présentant ce genre de temps d'attente auront toujours une qualité de service trop faible pour les passagers. Dans chacun des trois cas, le schéma du haut représente une situation avec des temps d'attentes absurdes représentés par des points de suspension, et le schéma du bas la situation correspondante sans ces temps d'attentes. Le cas du haut indique que les suppressions des attentes entre un ramassage et une livraison se font par retardement du ramassage et/ou avancement de la livraison suivante. Ceci traduit uniquement le fait que les gens n'attendent pas à l'arrêt juste après être monté ni juste avant de descendre quelque soit leur date de réalisation de service souhaitée. Les deux cas de figure en-dessous, illustrent la transitivité de ces temps d'attente absurdes entre services de même nature. On en déduit que seules les attentes dans le véhicule entre une livraison suivie d'un ramassage peuvent avoir du sens, ce qui se traduit par l'équation (3.6).

$$\forall (i, j) \notin \mathcal{R}^- \times \mathcal{R}^+, w_{ij} = 0 \quad (3.6)$$

Même si, à notre connaissance aucun outil d'optimisation n'en tient compte actuellement, ces temps d'attente ne sont pas si absurdes dans certains cas pratiques. Pour des raisons de sécurité par exemple, les passagers peuvent préférer attendre dans le véhicule avant de faire la correspondance avec un autre moyen de transport.

Nous signalons que bon nombre d'applications n'autorisent aucun temps d'attente à l'arrêt lorsque des passagers sont à bord. Ceci dans le souci d'augmenter le niveau de qualité de service ; ce qui paraît assez incohérent lorsque les passagers sont prêts à faire des détours.

3. *Ordre des services en un même lieu*

Dès qu'un véhicule arrive à un arrêt, il y effectue toutes ses livraisons sans attendre. Il quitte ce lieu dès que tous les ramassages prévus sont faits. Ceci est illustré par la figure 3.5 et s'exprime par la formule (3.7) :

$$\forall (v_i, v_j) \in \{\mathcal{A} \text{ avec } T_{ij} = 0\} \left\{ \begin{array}{l} t_j \geq t_i + s_i \quad \text{si } (i, j) \in \mathcal{R}^- \times \mathcal{R}^+ \text{ et } (v_i, v_j) \in \bar{\omega}; \\ t_j = t_i + s_i \quad \text{si } (i, j) \in \mathcal{R}^- \times \mathcal{R}^- \text{ et } (v_i, v_j) \in \bar{\omega}; \\ t_j = t_i + s_i \quad \text{si } (i, j) \in \mathcal{R}^+ \times \mathcal{R}^+ \text{ et } (v_i, v_j) \in \bar{\omega}; \\ (v_i, v_j) \notin \bar{\omega} \quad \text{si } (i, j) \in \mathcal{R}^+ \times \mathcal{R}^-. \end{array} \right. \quad (3.7)$$

On pourrait supposer que les livraisons et les ramassages sont simultanés et alors d'une durée égale au plus long des services ou à une autre combinaison. A priori, par souci d'homogénéité avec le cas général, nous considérons que ces temps de service se cumulent dans le temps. Ce qui nous permet d'intégrer les temps de service dans les temps de parcours.

4. *Différents niveaux de qualité de service par type de véhicule*

Les passagers peuvent exprimer des préférences (ou des exigences) par rapport au véhicule qui les transporte. Soit vec_r^k la préférence d'une requête r pour un type de véhicule k . Dans le cas où vec_r^k est binaire, chaque type de véhicule se voit attribuer un sous-graphe propre restreint à l'ensemble des requêtes qu'il peut servir.

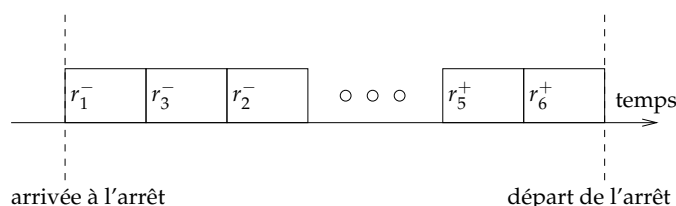


FIG. 3.5: Ordonnancement des services pour un véhicule en un même lieu entre deux trajets

5. Distance parcourue

Ce critère est le plus classique en calcul de tournées de véhicules. La distance est un attribut associable à chaque arc (tronçon) (v_i, v_j) . Que ce soit par course ou par tournée, il suffit de sommer la distance de chaque arc concerné. Malgré tout, il peut être pertinent de modéliser le réseau pour un p -graphe, rendant ainsi ce cas plus complexe. Ce sujet est développé dans la section 3.3.3.

6. Durée

C'est exactement le même calcul que celui de la distance parcourue en utilisant la durée des arcs et en y ajoutant les durées de service et les temps d'attente aux arrêts. Les seuls tronçons minimaux pour la durée sont suffisants pour obtenir une solution optimale. Si une contrainte est imposée sur la distance totale parcourue, il devient alors pertinent de considérer des trajets alternatifs comme pour le critère de la distance parcourue.

7. Pénibilité

À chaque tronçon de route, est associable un facteur de pénibilité calculé à partir du nombre de ralentissements ou d'arrêts obligatoires (feux tricolores, stops, passages piétons, ralentisseurs...), de la qualité du macadam, de la sinuosité de la route, de la beauté du paysage, ou tout autre critère déterminable de façon définitive et indépendante pour chaque tronçon.

3.2.2 Évaluation sur une course

1. Déroulement d'une course

Le temps qu'un passager va mettre pour se rendre de son lieu d'origine vers sa destination est un des premiers critères que l'on va chercher à minimiser. La figure 3.6 est une décomposition d'un voyage en ces différentes phases sur lesquelles peuvent porter l'optimisation. Il y a tout d'abord l'accès au service, puis le transport pris en charge par un véhicule (appelé course), puis le déplacement du point de livraison à la destination finale. Le tout pouvant être interrompu par des arrêts et des temps d'attente (représentés par des séries de trois cercles).

(a) Temps d'accès au service et/ou destination finale

Cela dépend directement de la distribution géographique des points de montée et de descente. Par exemple, ce temps est nul pour du porte à porte. Pour un semis d'arrêts donné, il s'agit d'affecter les passagers aux arrêts adéquats.

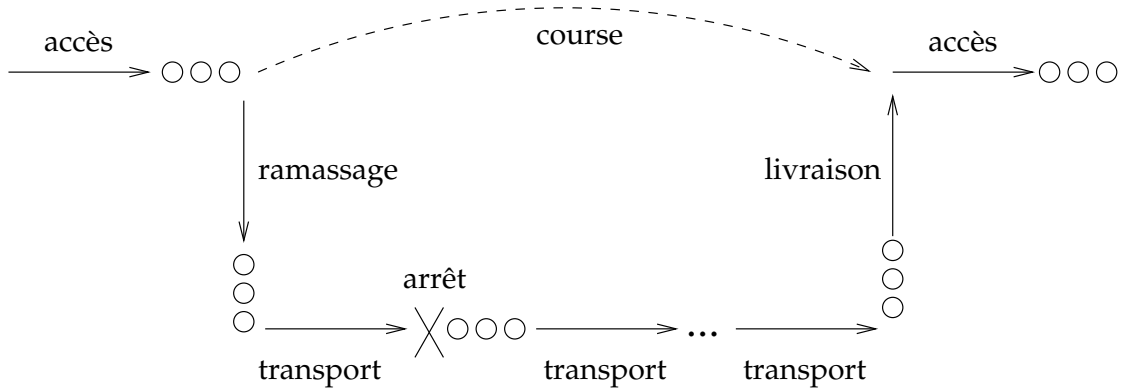


FIG. 3.6: Décomposition de la durée d'une course

Dans le DARP classique, ce critère est optimisé de façon indépendante ; c'est-à-dire que cette affectation est faite a priori (au plus proche). Intégrer ce critère dans l'optimisation du calcul des tournées, sous-entend de disposer d'un modèle distinguant l'origine et la destination des passagers de leurs points de ramassage et de livraison ainsi que de la possibilité d'affecter différents arrêts à un même service.

(b) Durée d'une course

Par définition, c'est le temps écoulé du début du ramassage jusqu'à la fin de la livraison. On note la durée de la course pour la requête r , Δ_r . Elle s'exprime suivant l'équation (3.8). Il est évident que lorsque les temps de service sont fixes, il est équivalent, en terme de minimisation, de considérer l'écart entre les dates de début de service.

$$\forall r \in \mathcal{R}, \Delta_r = s_{r^-} + t_{r^-} - t_{r^+} \quad (3.8)$$

Une course est composée d'une série de transports (roulants) interrompus par des arrêts pouvant générer des temps d'attente (figure 3.6). Ces attentes à l'arrêt sont souvent plus dérangeantes pour le passager que le temps passé à rouler. On peut chercher à limiter leur nombre, leur durée ou leur durée cumulée. Soit w_{ij} (donné par l'équation (3.2)) le temps d'attente entre deux services consécutifs i et j joints par l'arc (v_i, v_j) .

On peut introduire de la qualité de service en intervenant sur le nombre et/ou la durée de ces temps d'attente. Nous pouvons, par exemple, considérer la somme des durées de ces arrêts (W_r), dont voici l'expression (équation (3.9)).

$$\forall r \in \mathcal{R}, W_r = \sum_{(v_i, v_j) \in \bar{\omega}_{r^+, r^-}} w_{ij} \quad (3.9)$$

Dans l'équation 3.9, on reste dans le cas très général où les passagers peuvent attendre avant de quitter leur arrêt de montée et attendre avant de descendre à leur arrêt de destination. Dans le cas qui nous intéresse, on peut se restreindre à la sommation sur les couples $(i, j) \in \mathcal{R}^- \times \mathcal{R}^+$ comme cela est expliqué dans le paragraphe sur les temps d'attente absurdes.

2. *Pérégrinations*

On peut imposer une synchronisation à respecter entre deux requêtes différentes. Par exemple, on peut imposer un temps de latence, Lat_{ij} , minimal entre deux requêtes i et j pour un même usager, une pour l'aller, l'autre pour le retour. A priori, ces requêtes peuvent être prises en charge par différents véhicules. Il suffit pour cela de considérer la contrainte (3.10).

$$\forall (v_i, v_j) \in \mathcal{A}, t_{j^+} \geq t_{i^-} + s_{i^-} + Lat_{ij} \quad (3.10)$$

3. *Regroupement*

On peut aussi estimer le temps, pour un passager, passé seul dans le véhicule. Ceci peut-être positif ou négatif suivant les applications. La quantité $Meet_r$ (équation (3.12)) mesure tout au long de la course relative à la requête r le nombre de personnes – autres que les passagers de la requête r – dans le véhicule multiplié par le temps passé par ces personnes dans le véhicule. Pour cela, il faut déterminer l'ensemble $Vois(r)$ (équation (3.11)) des requêtes coïncidant avec la requête r .

$$Vois(r) = \left\{ r_2 \in \mathcal{R} \mid \{ \{ r_2^+, r_2^- \} \cap \dot{\omega}_{r^+ r^-} \} \cup \{ \{ r^+, r^- \} \cap \dot{\omega}_{r_2^+ r_2^-} \} \neq \emptyset \right\} \quad (3.11)$$

$$Meet_r = \sum_{r_2 \in Vois(r)} (\min(t_{r^-}, t_{r_2^-}) - \max(t_{r^+}, t_{r_2^+})) F_{r_2^+} \quad (3.12)$$

4. *Régularité des rencontres*

Une personne utilisant régulièrement le service peut espérer suivre des trajets similaires à chaque utilisation du service. Soit $aff_{r_1 r_2}$ la quantité d'affinité entre les passagers des requêtes r_1 et r_2 , l'équation (3.13) donne une mesure totale de l'affinité « reçue » par les passagers de la requête r_1 , notée Aff_{r_1} , en multipliant le temps passé par les passagers des requêtes coïncidentes avec r_1 (éléments de $Vois(r_1)$, défini au paragraphe sur le regroupement par leur quantité d'affinité avec r_1 .

$$Aff_{r_1} = \sum_{r_2 \in Vois(r_1)} (\min(t_{r_1^-}, t_{r_2^-}) - \max(t_{r_1^+}, t_{r_2^+})) aff_{r_1 r_2} \quad (3.13)$$

5. *Sinuosité*

La sinuosité d'un trajet peut-être vu comme une source d'inconvénient car les virages génèrent de l'inconfort et sont aussi associables à des détours. La sinuosité d'une courbe – assimilée à un trajet – peut-être mesurée de nombreuses façons. Nous reviendrons sur ce critère dans la section 6.2.

3.2.3 Évaluation sur une tournée

1. *Temps de repos*

Afin de respecter les conventions collectives et les réglementations autoroutières ou simplement d'améliorer les conditions de travail des chauffeurs, les tournées peuvent être soumises à certaines contraintes assurant des temps de repos aux

chauffeurs. Généralement, la durée totale d'une tournée attribuée à un chauffeur est limitée ainsi que les périodes de conduites sans pauses. La longueur des pauses pouvant dépendre de celle de la période roulée précédente.

Dans le cas pratique du Doubs Central que nous avons étudié, le transport est géré en sous-traitance par des compagnies de taxi qui poursuivent en parallèle leur propre activité. Les pauses idéales – sans passagers ($f_i = 0$) – doivent donc être soit très courtes ($\leq W^{bas}$), pour ne pas perdre de temps, soit très longues ($\geq W^{haut}$), pour permettre de prendre des passagers hors du TAD. Sous forme de contraintes, ceci se traduit par l'inégalité (3.14).

$$|w_{ij} - \frac{W^{bas} + W^{haut}}{2}| \geq \frac{W^{haut} - W^{bas}}{2} (1 - f_i) \delta_{ij} \quad (3.14)$$

3.2.4 Évaluation sur une solution

1. Taux de remplissage

Le taux de remplissage des véhicules est un critère qui outre le fait de participer à un aspect social du TAD, permet aussi d'en évaluer la performance et la pertinence sur un territoire. Il peut être intéressant de l'évaluer sur chaque tournée indépendamment si on considère des seuils d'acceptabilité sur le taux de remplissage. La formule (3.15) en donne une définition en fonction des durées de chaque tronçon et de la capacité du véhicule, qui fait le rapport entre la somme des produits de la durée de chaque tronçon avec son niveau d'occupation et la durée totale de la tournée, notée $duree(\omega)$ et égale à $t_{-(R+k)} - t_{R+k}$ si k est le véhicule servant ω .

$$Meet_{\omega} = \frac{\sum_{(v_i, v_j) \in \bar{\omega}} [(t_j - t_i) f_i / F]}{duree(\omega)} \quad (3.15)$$

Pour passer au taux de remplissage global sur plusieurs tournées, la fonction n'est pas linéaire. Le taux de remplissage global $Meet$ (équation (3.16)) est donné par le rapport entre la somme des numérateurs de chaque tournée et la somme des dénominateurs de chaque tournée. Nous notons Ω^* l'ensemble des tournées d'une même solution.

$$Meet = \frac{\sum_{\omega \in \Omega^*} \sum_{(v_i, v_j) \in \bar{\omega}} [(t_j - t_i) f_i / F]}{\sum_{\omega \in \Omega^*} duree(\omega)} \quad (3.16)$$

Il est aussi pertinent de calculer le taux de remplissage uniquement sur les portions roulantes des tournées. Les valeurs de $Meet'_{\omega}$ et de $Meet'$ sont respectivement données par les équations (3.17) et (3.18).

$$Meet'_{\omega} = \frac{\sum_{(v_i, v_j) \in \bar{\omega}} [T_{ij}(f_i \div F)]}{\sum_{(v_i, v_j) \in \bar{\omega}} T_{ij}} \quad (3.17)$$

$$Meet' = \frac{\sum_{\omega \in \Omega^*} \sum_{(v_i, v_j) \in \bar{\omega}} [T_{ij}(f_i \div F)]}{\sum_{\omega \in \Omega^*} \sum_{(v_i, v_j) \in \bar{\omega}} T_{ij}} \quad (3.18)$$

2. Sensibilité aux perturbations

Les perturbations dans un TAD sont du type modification du réseau, retard d'un véhicule ou d'un passager, modification de l'ensemble des requêtes à servir. . . La sensibilité du système à ces perturbations peut se mesurer de différentes façons : par la capacité à trouver une nouvelle solution acceptable en un temps raisonnable ; par l'écart entre l'ancienne solution et la nouvelle. La qualité de service est alors associée à des notions de robustesse, de la méthode de calcul des tournées comme de celle des solutions.

Une partie de la sensibilité des solutions est mesurable en contrôlant le niveau de saturation des fenêtres de temps allouées à chaque service. La formule (3.19) exprime l'écart par rapport aux limites des fenêtres de temps pour les ramassages et les livraisons.

$$\Delta_i(T) = \begin{cases} t_i - T_i^{inf} & \text{si } i \in \mathcal{R}^+ \\ T_i^{sup} - t_i & \text{si } i \in \mathcal{R}^- \end{cases} \quad (3.19)$$

En effet, une tournée dont la somme des $\Delta_i(T)$ est important est plus apte à accepter de nouvelles requêtes. Une telle mesure est aussi applicable sur d'autres contraintes comme celle liée à la capacité du véhicule.

Un critère permettant de diminuer la sensibilité du système à l'arrivée de nouvelles requêtes, est la répartition spatio-temporelle des véhicules. Si à chaque instant l'ensemble des véhicules sont répartis de façon rationnelle face à la distribution de l'apparition de nouvelles requêtes sur le territoire, alors le système est apte à accueillir des nouvelles requêtes.

3. Forme des tournées

Plusieurs aspects liés à la forme dans l'espace et dans le temps des tournées peuvent nuire à la qualité de service offerte aux usagers. Là aussi certains des critères sur la forme peuvent s'évaluer à différents niveaux mais nous estimons qu'il n'est pas raisonnable de ne s'intéresser que partiellement aux problèmes liés à la forme des tournées. Passer deux fois au même endroit, passer proche de la destination finale, effectuer des boucles, ou privilégier (ou éviter) des tronçons particuliers sont des caractéristiques auxquelles les passagers et les transporteurs peuvent accorder de l'importance. La figure 3.7 (en haut) présente un cas de solution « indésirable ». En effet, les passagers « hachurés en oblique » passent proche de leur destination puis s'en éloignent. Plusieurs approches peuvent permettre d'éviter ce genre de problème. Dans le premier cas (au milieu à gauche), on choisit un arc reliant les deux mêmes services correspondant à un meilleur trajet sur le réseau routier. Ceci implique de connaître dans le modèle, le réseau et tous les chemins pouvant mener d'un arrêt à un autre. Le nombre de données et leur traitement devient alors plus complexe que dans le cadre classique où on se contente d'un modèle basé sur un graphe simple.

Dans le deuxième cas (au milieu à droite), on choisit un arrêt plus avantageux pour le même service. Ceci sous-entend que pour chaque service, on dispose d'un ensemble d'arrêts possibles. Là aussi on augmente fortement l'aspect combinatoire du problème.

Dans le troisième cas (en bas), on modifie la séquence.

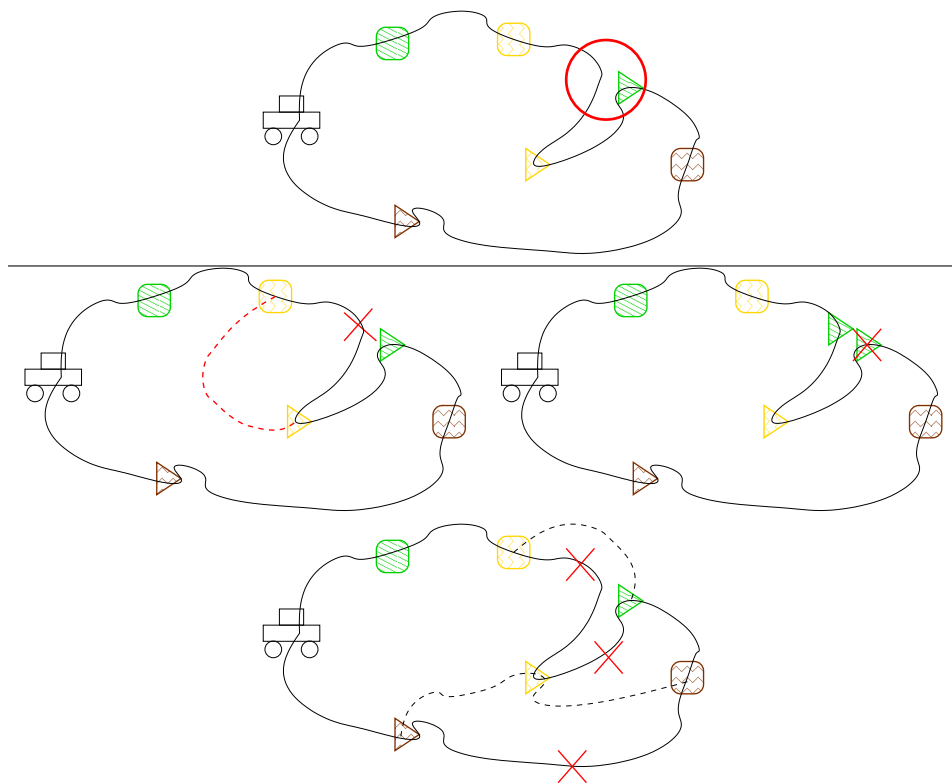


FIG. 3.7: Modification de la course à partir du réseau

Il paraît peu pertinent de ne pas considérer toutes les approches citées pour éviter des formes de tournées « indésirables ». Car modifier une séquence alors qu'il suffit de déposer les passagers à un autre endroit ou d'effectuer un léger détour, serait absurde d'un point de vue applicatif. La prise en compte de ce critère de qualité de service nécessite donc une modélisation du problème complexe et un volume de données important.

On peut néanmoins aisément en tenir compte en post-traitement, une fois les tournées calculées, et en disposant de toutes les données géographiques.

Dans la suite nous considérons que \mathcal{A} contient tous les trajets offerts par le réseau routier entre deux arrêts et qu'il est possible de réaliser un même service en différents arrêts. Nous notons $(v_i, v_j)^e$ le $e^{\text{ième}}$ arc de v_i vers v_j et $Arr(i)$ l'ensemble des sommets possibles pour i .

(a) Forme d'une course

Pour un passager, les critères sur la forme ont un impact sur la façon dont il perçoit le service plus que sur le niveau réel de qualité de service qui lui est offert. Par exemple, un passager peut mal percevoir un détour qui lui paraît inutile même s'il permet un service dans de meilleurs délais. Il paraît concevable de faire passer un passager deux fois au même endroit ; c'est à dire faire des boucles (figure 3.8 à gauche, la requête hachurée en oblique). A la nuance près qu'un passager, une fois sa course entamée, ne comprendrait pas de passer à proximité de son arrêt de ramassage ou de son arrêt de livraison sans être déposé (figure 3.8 à droite, requête hachurée en oblique).

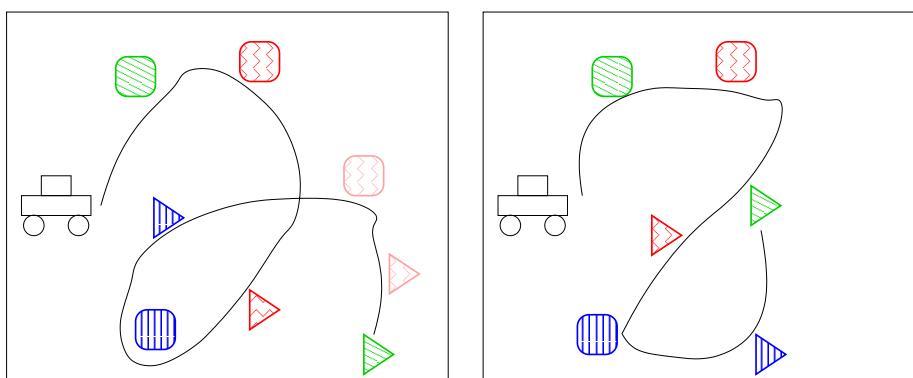


FIG. 3.8: Deux types de boucles

D'après ces remarques, on peut envisager d'interdire (ou de pénaliser) l'utilisation de certains trajets pendant une course. La détermination du niveau de gêne, noté $Prox_r((i, j)^e)$, lié au passage par un trajet (arc) $(v_i, v_j)^e$ à proximité d'un arrêt extrême d'une requête r au cours d'une course, peut être effectuée en prétraitement. Le problème de ce type de mesure est que les arcs a priori les plus intéressants pour servir la requête ont une valeur de $Prox_r((i, j)^e)$ très élevée – car ils passent par définition à proximité de r^- . Le cas de valeurs binaires pour mesurer l'inconvénient permet de prendre en compte ce

critère assez simplement. La contrainte (3.20) assure que le véhicule ne passe qu'une fois à proximité du lieu de livraison de la requête r après en avoir effectué le ramassage.

$$\forall r \in \mathcal{R}, |\{(v_i, v_j)^e \in \bar{\omega}_{r+r^-} | Prox_r((i, j)^e) = 1\}| = 1 \quad (3.20)$$

(b) Forme d'une tournée

Lorsqu'un opérateur utilise un logiciel calculant des tournées de véhicules, certaines formes de tournées lui paraissent plus « optimales » que d'autres même si elles sont moins performantes sur les critères d'optimisation qu'il fournit. En particulier, les tournées effectuant peu de croisement sont privilégiées (Lu et Dessouky, 2006). Dans l'exemple de la figure 3.9 l'opérateur préfère la tournée de droite même si son coût est supérieur. On peut ainsi espérer qu'un système automatisé soit mieux accepté.

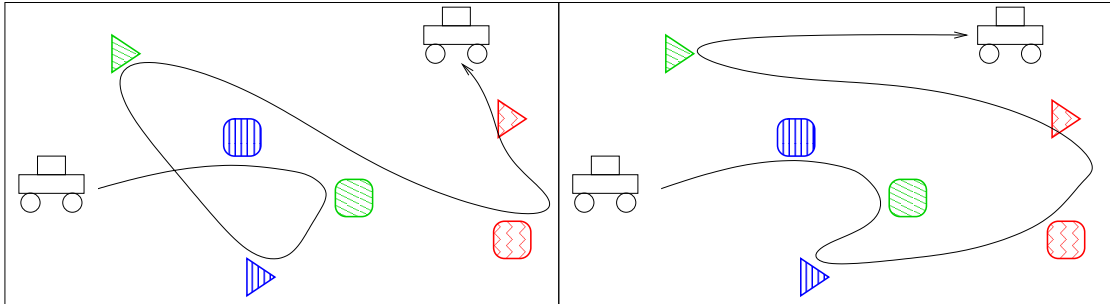


FIG. 3.9: Tournées avec et sans croisements

(c) Forme d'une solution

Les tournées effectuées par l'ensemble des véhicules peuvent également causer du désagrément aux usagers, si par exemple l'un d'eux passe à leur arrêt (ou à proximité) sans les ramasser. La figure 3.10 illustre cette situation. Dans ce cas, le véhicule de gauche ramasse les passagers de la requête hachurée horizontalement, sans prendre ceux de la requête hachurée verticalement, ou inversement avec le véhicule de droite. Ce critère est particulièrement difficile à prendre en compte puisqu'il met en jeu toutes les tournées dans la dimension spatiale (proximité) et temporelle (passer peu avant un ramassage à proximité de son arrêt).

4. Nombre de véhicules

Souvent la limite sur le nombre de véhicules disponibles n'est pas une contrainte forte puisque l'on suppose que le TAD est prévu avec un nombre suffisant de véhicules pour servir toutes les requêtes. Un coût fixe, intégré à la fonction objectif, est souvent lié à l'activation d'un véhicule. Dans plusieurs applications, la minimisation du nombre de véhicules utilisés est un objectif prioritaire. À ce moment là, une approche couramment utilisée est d'attribuer une forte pénalité à

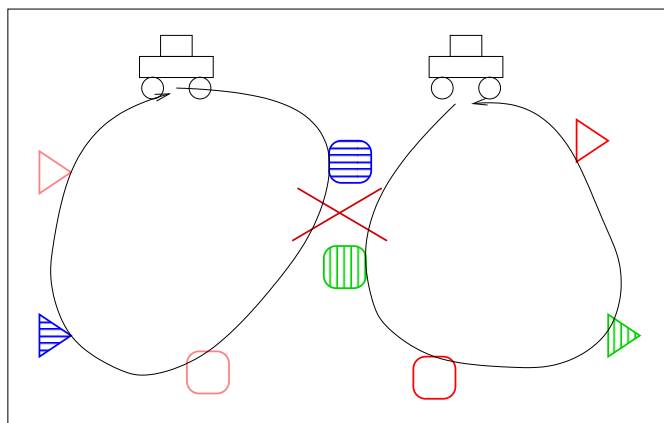


FIG. 3.10: Deux tournées en conflit

l'activation de chaque véhicule. Dans ce cas, Desaulniers (2007) montre qu'une gestion dynamique incrémentale de ces pénalités est préférable en terme de performances.

5. *Équilibrage des charges*

La charge de travail est répartie de façon équitable entre les différents chauffeurs ou compagnies de chauffeurs. Cela peut se faire par rapport au nombre de requêtes servies ou à la durée des tournées.

6. *Différenciation des passagers*

Les collectivités locales peuvent choisir de privilégier certains usagers, par exemple parce que leur zone d'habitation est moins bien desservie par d'autres modes de transport ou pour des critères sociaux. Un coefficient propre à chaque type d'utilisateur peut être intégré à la fonction objectif.

3.2.5 Évaluation à long terme

1. *Temps d'obtention de la réponse du système et délai pour l'acceptation d'une nouvelle requête*

Plus le système est capable de fournir une réponse rapidement, plus cela accorde de la souplesse au client. Ce critère prend toute son importance dans le cas dynamique, puisqu'il permet de déterminer jusqu'à combien de temps à l'avance un usager peut émettre une nouvelle requête. À cela on peut associer un pourcentage d'acceptation dépendant de la robustesse des solutions trouvées. La qualité de service est d'autant plus élevée que la probabilité d'acceptation d'une requête $p(\epsilon)$ au temps $t - \epsilon$ est inversement proportionnelle à la valeur de ϵ .

Avant d'envisager l'intégration de ces critères de qualité de service dans le problème général de calcul de tournées, nous envisageons l'optimisation de certains d'entre eux dans le cas d'une optimisation sur une séquence fixée d'arrêts à visiter.

3.3 Maximisation de la qualité de service pour une séquence fixée d'arrêts à visiter

3.3.1 Définition et modélisation

Que ce soit dans des heuristiques ou des méthodes exactes, le DARP est souvent décomposé en sous-problèmes. Les premières étapes consistent à déterminer quels véhicules – et dans quel ordre – vont effectuer les services. Une fois de telles séquences fixées l'optimisation est rarement finie. Dans un cas classique la séquence de sommets fixée implique la détermination exacte de la séquence d'arcs – trajets – à emprunter. Deux cas peuvent alors se présenter : la fonction objectif dépend des heures de réalisation des services ou la fonction objectif n'en dépend pas. Même dans le deuxième cas, la plupart des auteurs résolvent le problème d'horodatage de la séquence en passant par l'utilisation d'une fonction objectif secondaire basée sur les dates de réalisation des services. Un cas plus original est la considération de différents arcs entre chaque paire de sommets, l'optimisation doit alors non seulement sélectionner les dates de réalisation des services mais aussi les arcs à emprunter.

Nous limitons l'horodatage à la détermination des dates de service. Nous n'intégrons pas les heures de départ du véhicule aux arrêts, nous estimons que celles-ci se déduisent trivialement une fois les dates de service fixées.

3.3.2 Cas du 1-graphe

3.3.2.1 Modélisation

Soit $G_\omega = (\mathcal{V}_\omega, \mathcal{A}_\omega)$ un 1-graphe, où \mathcal{V}_ω est la restriction de \mathcal{V} à la séquence $\bar{\omega}$ et \mathcal{A}_ω les arcs la composant. Le problème d'horodatage HP prend la forme suivante. La fonction ϕ de $\mathbb{R}^{|\mathcal{V}_\omega|}$ dans \mathbb{R} exprime l'inconvénient lié aux dates de réalisation des services. Les contraintes (3.22) assurent le respect des fenêtres de temps tandis que les contraintes (3.23) assurent la continuité du flot sur la ressource T .

$$(HP) \quad \min \phi(t) \quad (3.21)$$

s.c.q.

$$T_i^{inf} \leq t_i \leq T_i^{sup} \quad \forall v_i \in \mathcal{V}_\omega, \quad (3.22)$$

$$t_i + T_{ij} - t_j \leq 0 \quad \forall (v_i, v_j) \in \mathcal{A}_\omega, \quad (3.23)$$

$$t \in \mathbb{R}^{|\mathcal{V}_\omega|} \quad (3.24)$$

Le fait d'imposer un ordre sur les services, permet de renforcer les contraintes de fenêtres de temps $[T_i^{inf}, T_i^{sup}]$ initiales. Il est clair que certaines dates de début de service

peuvent être incompatibles avec des services prédécesseurs ou successeurs. Les procédures récursives, où \underline{i} représente le successeur de i dans $\bar{\omega}$, décrites par les équations (3.25) et (3.26) permettent la mise à jour des fenêtres de temps.

$$T_{\underline{i}}^{inf} := \max \left\{ T_i^{inf} + T_{i\underline{i}}, T_{\underline{i}}^{inf} \right\} \quad \forall v_i \in \mathcal{V}_\omega \quad (3.25)$$

$$T_i^{sup} := \min \left\{ T_{\underline{i}}^{sup} - T_{i\underline{i}}, T_i^{sup} \right\} \quad \forall v_i \in \mathcal{V}_\omega \quad (3.26)$$

La séquence est supposée valide ; c'est-à-dire que toutes les contraintes structurelles (élémentarité, couplage, précédence) et d'occupation du véhicule sont respectées et qu'il existe un horodatage réalisable pour HP.

3.3.2.2 Résolution

Lorsque les dates de service peuvent être considérées à valeurs réelles (ce qui, en pratique, est généralement le cas), HP est un programme linéaire si ϕ est linéaire et sinon sa difficulté dépend de la nature de ϕ et fait appel à des méthodes de résolution plus ou moins standard. La littérature propose néanmoins des algorithmes *ad hoc*.

Le cas des fonctions objectifs exprimables comme combinaison linéaire de fonctions d'inconvénient propres à chaque service représente la quasi-totalité de la littérature à ce sujet (Desrosiers et al., 1995). Nous remarquons que sous ces conditions, le problème est analogue à un problème d'ordonnancement à une machine de type $(1, F_k | (r_j, d_j), seq | line)$, où 1 est le nombre de machines, F_k la limite de ressource consommée, r_j et d_j les dates de réalisation au plus tôt et au plus tard, *seq* (pour séquence) le type de graphe représentant les contraintes de précédence et *line* (pour linéaire) le type de métrique. Puisque les durées de transport entre deux services successifs sont constants, on peut appeler ces trajets des tâches de durée fixe. Les contraintes de précédence entre les tâches définissent l'ordre total équivalent à la séquence de services donnée. Le but est de déterminer les dates de début pour une chaîne de tâches de durées fixes. La tâche i correspond au trajet entre les services i et \underline{i} . Le début d'une tâche i correspond donc au démarrage du service i . Considérons une fonction de pénalité (3.27) pour chaque tâche linéaire en trois morceaux.

$$\phi_i = \max \left\{ \alpha_i (T_i^{inf} - t_i), 0, \beta_i (t_i - T_i^{sup}) \right\} \quad (3.27)$$

La pénalité pour toute la chaîne s'obtient en sommant ces valeurs. Garey et al. (1988) proposent un algorithme en $\mathcal{O}(n \log n)$ résolvant le problème pour $\alpha_i = \beta_i = 1$, où n est le nombre de tâches. Hoogeveen et van de Velde (1996) montrent que cet algorithme fonctionne pour $\alpha_i = \alpha$ et $\beta_i = \beta$. Puis Chrétienne (1999) propose un algorithme toujours en $\mathcal{O}(n \log n)$ pour le cas avec des coefficients de pénalité asymétriques et indépendants des tâches qui s'étend au cas des coefficients dépendants des tâches (Chrétienne et Sourd, 2000). Pour le cas général, Szwarc et Mukhopadhyay (1995) donnent un algorithme en $\mathcal{O}(nm)$ où m est le nombre de groupes (dans la solution) qui sont des paquets de tâches se succédant sans temps d'attente.

Nous présentons quelques critères et algorithmes de résolution adéquats.

3.3. Maximisation de la qualité de service pour une séquence fixée d'arrêts à visiter

1. *La date de fin de la tournée*

Dans le cas où l'objectif est de minimiser la date de fin de la tournée, on a la solution optimale triviale telle que $t_i = T_i^{inf}, \forall v_i \in \mathcal{V}_\omega$.

2. *Le temps perdu*

Si on ne considère que des requêtes *aller* (voir la section 2), la fonction d'inconvénient relative à la date de ramassage de chaque passager est linéaire décroissante. Optimiser suivant ce critère revient à réaliser un horodatage au plus tard. L'inconvénient mesuré étant uniquement localisé aux ramassages, on peut utiliser un second critère hiérarchique sur les dates de livraison. Ces dernières sont généralement prises au plus tôt afin de minimiser les durées de transport et les temps d'attente mais surtout d'éviter les temps d'attente absurdes. Les requêtes *retour* offrent un raisonnement symétrique avec les dates de livraison au plus tôt. Le cas du mélange des deux types de requêtes est moins trivial et peut être vu comme un cas particulier « d'une combinaison linéaire de fonctions convexes d'inconvénient par rapport à une date de réalisation de service » présenté par la suite.

3. *Une combinaison linéaire de l'écart entre la durée effective de la course et sa durée minimale et de l'écart entre la date de service et une date limite de réalisation*

Le problème esclave de la décomposition de Benders de [Sexton et Bodin \(1985a\)](#) pour le 1-DARP, est un problème d'horodatage optimisé suivant la fonction objectif (3.28).

$$\phi = \min \sum_{r \in \mathcal{R}_\omega} [\beta(t_{r-} - t_{r+}) - \alpha t_{r-}] \tag{3.28}$$

Cette fonction d'inconvénient se ramène aisément à une fonction linéaire en chaque service (voir figure (3.11)). Ils proposent deux algorithmes résolvant le problème

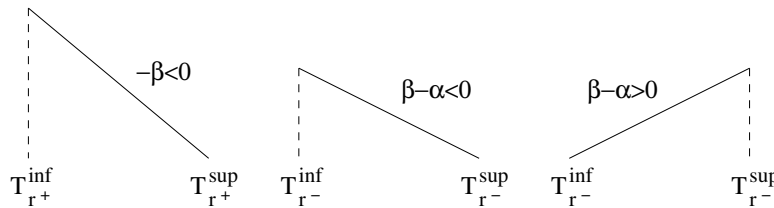


FIG. 3.11: Fonction d'inconvénient liée aux dates de service (Sexton et al.).

de flot équivalent au dual du problème d'horodatage. Le premier algorithme répond au cas $\beta > \alpha$ et le second au cas $\beta \leq \alpha$. Ces algorithmes sont de complexité linéaire. [Sexton et Choi \(1986\)](#) généralisent ϕ par une combinaison de fonctions linéaires en trois morceaux pour mesurer le désagrément en chaque service, autorisant ainsi les retards de livraison.

4. *Une combinaison linéaire de fonctions convexes d'inconvénient par rapport à une date de réalisation de service*

[Dumas et al. \(1990\)](#) donnent un algorithme basé sur la résolution séquentielle de minimisations de problèmes d'horodatage ayant les contraintes de succession

(3.22) relâchées. Ces dernières étant intégrées au fur et à mesure. Cette modélisation, figure 3.12, est assez proche de l'inconvénient subi par les passagers. Dans le cas particulier de fonctions quadratiques, l'algorithme est montré linéaire. Bien sûr cet algorithme offre une alternative à celui de [Sexton et Bodin \(1985a\)](#) dans le cas de fonctions d'inconvénients linéaires. Les auteurs montrent qu'il est aussi possible d'intégrer une fonction d'inconvénient linéaire liée aux temps d'attente et que l'algorithme reste valide dans le cas de date de services à valeurs entières. Ce sont sensiblement les mêmes algorithmes pour [Dumas et al. \(1990\)](#) et [Chrétienne et Sourd \(2000\)](#).

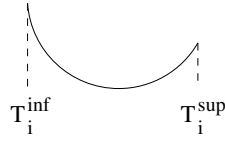


FIG. 3.12: Fonction d'inconvénient liée aux dates de service (Dumas et al.).

Un bloc est une suite de trajets consécutifs sans temps d'attente. On parcourt la séquence des trajets (entre deux arrêts successifs) de façon incrémentale. Pour un trajet de i à \underline{i} , soit on peut le positionner de façon à minimiser f_i et on passe au trajet suivant, sinon on réoptimise le bloc précédent. On réitère la réoptimisation des blocs tant que l'on rentre en conflit avec le bloc précédent, puis on passe au trajet suivant.

5. Une combinaison linéaire d'une déviation par rapport à une date de service souhaitée avec les temps d'attente et la durée individuelle de transport

[Ahuja et al. \(2003\)](#) traitent une telle fonction objectif (équation (3.29)), par transformation puis résolution en un problème dual de flot à coûts entiers et à objectif convexe (*The Dual Network Flow Problem*), modélisé par DNFP.

$$\min \sum_{v_i \in \mathcal{V}_\omega} [\alpha_i(t_i - T_i^{id}) + \beta_i(t_{\underline{i}} - T_{\underline{i}} - t_i)] + \sum_{v_{i+} \in \mathcal{V}_\omega} \gamma_{i+}(t_{i-} - t_{i+}) \quad (3.29)$$

(DNFP)

$$\min \sum_{(i,j) \in Q} \bar{H}_{ij}(w_{ij}) + \sum_{i \in P} \bar{B}_i(\mu_i) \quad (3.30)$$

s.c.q.

$$\mu_i - \mu_j \leq w_{ij} \quad \forall (i,j) \in Q, \quad (3.31)$$

$$l_{ij} \leq w_{ij} \leq u_{ij} \quad \forall (i,j) \in Q, \quad (3.32)$$

$$l_i \leq \mu_i \leq u_i \quad \forall i \in P, \quad (3.33)$$

$$w_{ij} \in \mathbb{N}, \mu_i \in \mathbb{N}, P \subset \mathbb{N}, Q \subset P \times P. \quad (3.34)$$

Les variables de décisions sont les w_{ij} et les μ_i . Les données l_i, u_i, l_{ij}, u_{ij} sont entières (l_i et l_{ij} peuvent prendre des valeurs négatives). \bar{H}_{ij} et \bar{B}_i sont des fonctions convexes. De plus l'ensemble $U = \max[\max \{u_{ij} - l_{ij} : (i,j) \in Q\}, \max \{u_i - l_i : i \in P\}]$ est fini.

3.3. Maximisation de la qualité de service pour une séquence fixée d'arrêts à visiter

En prenant $Q = \mathcal{A}_\omega$ et $P = \mathcal{V}_\omega$, Des changements de variable décrits par les équations (3.35)-(3.36) on déduit les valeurs suivantes aux données de DNFP (équations (3.37)-(3.40)).

$$w_{ij} = t_j - t_i - T_{ij}, \forall (v_i, v_j) \in \mathcal{A}_\omega \quad (3.35)$$

$$\mu_i = t_i, \forall v_i \in \mathcal{V}_\omega \quad (3.36)$$

De l'équation (3.35), on déduit aisément les contraintes (3.31).

$$l_{ij} = T_j^{inf} - T_i^{sup} - T_{ij}, \forall (v_i, v_j) \in \mathcal{A}_\omega \quad (3.37)$$

$$u_{ij} = T_j^{sup} - T_i^{inf} - T_{ij}, \forall (v_i, v_j) \in \mathcal{A}_\omega \quad (3.38)$$

Les deux équations (3.37) et (3.38) permettent, avec les contraintes (3.22) et (3.23) de HP, de déduire les contraintes (3.32) de DNFP.

$$u_i = T_i^{sup}, \forall v_i \in \mathcal{V}_\omega \quad (3.39)$$

$$l_i = T_i^{inf}, \forall v_i \in \mathcal{V}_\omega \quad (3.40)$$

Des contraintes (3.22) de HP et des équations (3.39) et (3.40), on déduit les contraintes (3.33) dans DNFP. Il est aisé de voir que la fonction objectif (3.29) sur les trois critères : la date de service, le temps d'attente et la durée de transport, se ramène à deux fonctions convexes du type de \bar{H}_{ij} et \bar{B}_i .

Le problème DNFP subit ensuite une suite de transformations et est résolu comme un problème de flot à coûts non linéaires.

6. Une somme de fonctions linéaires par morceaux d'inconvénient sur des contraintes souples de fenêtres de temps et de durée totale de transport

Le problème d'horodatage prend alors une forme différente (HPS) où ϕ_i^T mesure l'inconvénient lié aux dates de service et ϕ_{ij}^Δ celui lié à la durée de transport. La difficulté de HPS réside dans les contraintes d'intégralité (3.43) imposées sur les dates de service.

(HPS)

$$\min \sum_{v_i \in \mathcal{V}_\omega} \phi_i^T(t_i) + \sum_{(v_i, v_j) \in \mathcal{A}_\omega} \phi_{ij}^\Delta(t_j - t_i) \quad (3.41)$$

s.c.q.

$$t_i + T_{ij} - t_j \leq 0 \quad \forall (v_i, v_j) \in \mathcal{A}_\omega, \quad (3.42)$$

$$t_i \in \mathbb{N} \quad \forall v_i \in \mathcal{V}. \quad (3.43)$$

Hashimoto et al. (2006) prouvent par une réduction du problème de sac-à-dos, que le problème d'horodatage avec des contraintes souples de fenêtres de temps et de durée de transport ayant la forme de fonctions linéaires par morceaux, est pseudo-polynomial. De plus, si la contrainte sur la durée de transport est convexe le problème est alors polynomial. Ils fournissent un algorithme de programmation dynamique dans les deux cas.

3.3.3 Cas du p -graphe

La présence de contraintes de fenêtres de temps peut amener à considérer des tronçons alternatifs aux plus courts en distance entre chaque paire d'arrêts. En effet, un ensemble de tournées structurellement différent utilisant des tronçons parfois plus coûteux en distance mais plus rapides peut former une solution finalement moins coûteuse en distance.

La figure 3.13 illustre un exemple à deux requêtes pour lequel l'utilisation de tronçons alternatifs est justifiée. Il apparaît que la contrainte sur la date de livraison au plus

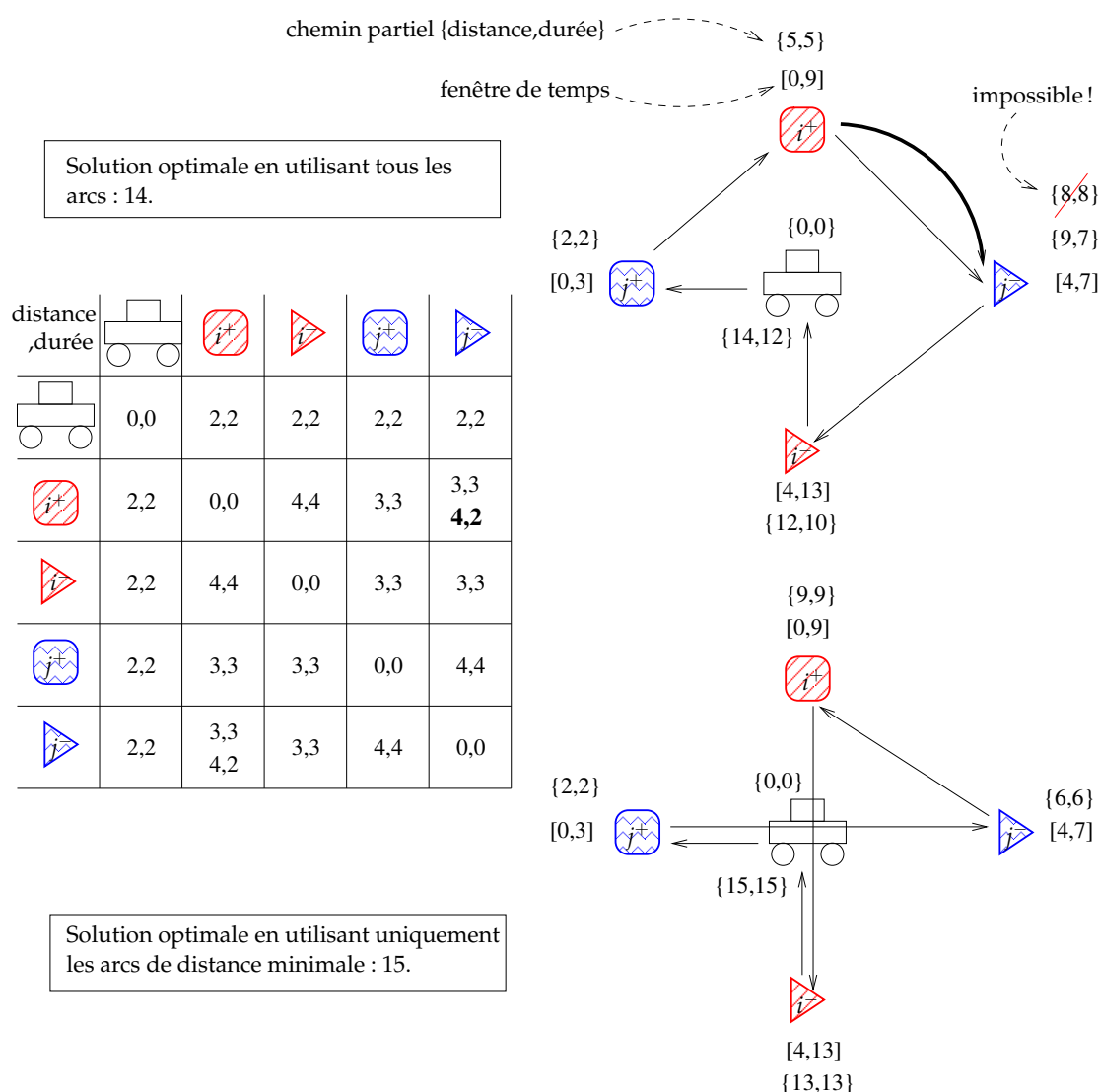


FIG. 3.13: Utilisation de chemins alternatifs

tard de la requête j (lignes brisées) ayant pour valeur 7, oblige l'utilisation de l'arc de distance maximale (4,2) entre le ramassage de i (hachuré) et la livraison de j , afin de

valider une tournée optimale. Il est évident que porter la valeur de cette contrainte à 8 permet d'obtenir une solution de moindre distance en utilisant l'arc de pondération (3,3) entre le ramassage i^+ et i^- .

Entre chaque paire de sommets, on considère donc un ensemble de tronçons offrant un compromis entre distance parcourue et durée de parcours ce qui se modélise par un p -graphe. Soit $G_\omega = (\mathcal{V}_\omega, \mathcal{A}_\omega)$ un p -graphe, où \mathcal{V}_ω est la restriction de \mathcal{V} à la séquence $\bar{\omega}$ et \mathcal{A}_ω les arcs la composant. Par définition, jusqu'à p arcs peuvent relier chaque paire de sommets. Pour que l'utilisation d'un p -graphe soit utile, il faut que les arcs entre chaque paire de sommets présentent un compromis intéressant. Dans cette section, nous prenons comme cas d'école celui de la minimisation de la distance totale parcourue (voir le paragraphe 5 de la section 3.2). Les arcs offrent des compromis sur la distance pour minimiser le coût et la durée pour satisfaire les contraintes de fenêtres de temps. Les critères de qualité de service uniquement basés sur des dates de réalisation de service ne nécessitent généralement pas l'utilisation d'un p -graphe ; les plus courts chemins en temps étant a priori dominants.

3.3.3.1 Modélisation

On note $(v_i, v_j)^e$ le $e^{\text{ième}}$ arc entre i et j offrant un compromis, par exemple durée/coût. Sa durée est notée T_{ije} et son coût C_{ije} . Par convention $(v_i, v_j)^0$ est l'arc de durée minimale. (HP) devient alors le problème en variables binaires suivant.

(HP)

$$\min \sum_{v_i \in \mathcal{V}_\omega \setminus \{v_{-(R+k)}\}} \sum_{e=1}^{|\{(v_i, v_j)^e \in \mathcal{A}_\omega\}|} C_{ije} \delta_{ije} \quad (3.44)$$

s.c.q.

$$T_i^{inf} \leq t_i \leq T_i^{sup} \quad \forall v_i \in \mathcal{V}_\omega, \quad (3.45)$$

$$\delta_{ije}(t_i + T_{ije} - t_j) \leq 0 \quad \forall (v_i, v_j)^e \in \mathcal{A}_\omega, \quad (3.46)$$

$$\sum_{(v_i, v_j)^e \in \mathcal{A}_\omega} \delta_{ije} \in \{0, 1\} \quad \forall \{v_i, v_j\} \in \mathcal{V}_\omega \times \mathcal{V}_\omega, \quad (3.47)$$

$$\delta_{ije} \in \{0, 1\} \quad \forall (v_i, v_j)^e \in \mathcal{A}_\omega. \quad (3.48)$$

Ce problème est un cas particulier à deux ressources et à borne de consommation unique ($T_i^{inf} = -\infty, T_i^{sup} = T, \forall i \in \mathcal{V}_\omega$) du problème que nous appelons Problème de Sélection des Arcs dans une Séquence Fixée (PSASF) ou *Fixed Sequence Arc Selection Problem* en anglais. Dans le cas d'une fonction objectif du type somme des coûts des arcs, (PSASF) est un problème NP-difficile car équivalent à un Problème de Sac-à-Dos Multidimensionnel à Choix Multiple (PSMCM), une extension NP-difficile du problème de sac-à-dos (Kellerer et al., 2004) qui est modélisé par les équations (3.49)-(3.52). En effet, si on considère que les objets sont les arcs, les dimensions sont les ressources et les classes d'objets sont les paires de sommets successifs. En notant i le successeur de i

dans la séquence, on définit $|v_i \in \mathcal{V}| - 1$ classes d'objets. Chaque objet e de la classe i est caractérisé par un poids T_{iie} et un coût C_{iie} . T est la limite de poids total imposée.

(PSASF)

$$\min \sum_{v_i \in \mathcal{V}_\omega \setminus \{v_{-(R+k)}\}} \sum_{e=1}^{|\{(v_i, v_i)^e \in \mathcal{A}_\omega\}|} C_{iie} \delta_{iie} \quad (3.49)$$

s.c.q.

$$\sum_{v_i \in \mathcal{V}_\omega \setminus \{v_{-(R+k)}\}} \sum_{e=1}^{|\{(v_i, v_i)^e \in \mathcal{A}_\omega\}|} T_{iie} \delta_{iie} \leq T \quad , \quad (3.50)$$

$$\sum_{e=1}^{|\{(v_i, v_i)^e \in \mathcal{A}_\omega\}|} \delta_{iie} = 1 \quad \forall i \in \mathcal{V}_\omega, \quad (3.51)$$

$$\delta_{iie} \in \{0, 1\} \quad \forall (v_i, v_i)^e \in \mathcal{A}_\omega. \quad (3.52)$$

3.3.3.2 Résolution

(HP) se formule comme un SPPRC (Problème de Plus Court Chemin avec Contraintes de Ressources) sur un graphe linéaire¹. Nous proposons de résoudre ce problème par une adaptation (algorithme 2) de l'algorithme de programmation dynamique 1 présenté dans la section 2.3.2.

La différence avec l'algorithme général est que le graphe peut être parcouru dans l'ordre de la séquence en étant sûr de ne pas passer deux fois par le même sommet. Les contraintes structurelles étant déjà respectées, la validité d'un chemin partiel ne nécessite que le contrôle des fenêtres de temps. Tout arc $(v_i, v_j)^e$ sert à étendre tous les chemins partiels entrant en i , puis est comparé avec tous les chemins partiels sortants de j déjà créés. On en déduit une complexité en $\mathcal{O}(|\mathcal{A}_\omega| p^2)$. Le nombre maximal de chemins partiels entre chaque paire de sommets, p est borné par le produit des nombres de valeurs possibles pour les durées et les coûts. L'algorithme 2 s'étend facilement au cas à plusieurs ressources (Garaix et al., 2007). La figure 3.14 illustre le déroulement de l'algorithme sur un graphe à trois sommets avec C représentant la distance et T la durée.

¹Nous appelons graphe linéaire tout graphe connexe dont tous les sommets ont exactement un successeur sauf un qui n'en a pas.

3.3. Maximisation de la qualité de service pour une séquence fixée d'arrêts à visiter

Algorithme 2 : Algorithme de résolution de (HP) sur un p -graphe

Données : G_ω un graphe linéaire

Résultat : $CH_{-(R+k)}$ les chemins arrivant au dépôt

```

1 Initialisation :  $CH_{R+k} :=$  un chemin réduit au sommet de départ;
2  $i := R + k$ ;
3 tant que  $i \neq -(R + k)$  faire
4   pour chaque chemin partiel  $ch \in CH_i$  faire
5     pour chaque arc  $(v_i, v_i)^e$  faire
6        $ch' :=$  l'extension de  $ch$  par  $(v_i, v_i)^e$ ;
7        $domine = faux$ ;
8       si  $ch'$  est valide alors
9         pour chaque chemin partiel  $ch'' \in CH_i$  faire
10          si  $T_{ch'} \leq T_{ch''}$  et  $C_{ch'} \leq C_{ch''}$  alors
11             $CH_i := CH_i \setminus \{ch''\}$ ;
12          sinon
13            si  $T_{ch''} \leq T_{ch'}$  et  $C_{ch''} \leq C_{ch'}$  alors
14               $domine := vrai$ ;
15              stop;
16          si  $domine = faux$  alors
17             $CH_i := CH_i \cup \{ch'\}$ ;

```

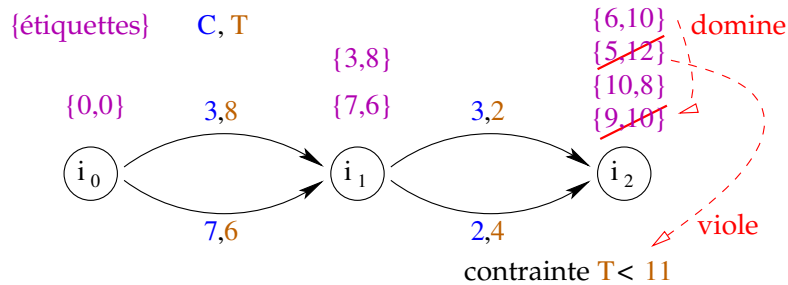


FIG. 3.14: Algorithme de programmation dynamique pour le Problème de Sélection des Arcs dans une Séquence Fixée

Chapitre 4

Maximisation de la qualité de service

RÉSUMÉ : Dans la section 4.1, nous proposons une méthode de génération de colonnes pour le problème de maximisation de la qualité de service présenté dans la section 3.1. Nous présentons les détails d'implémentation de la résolution des problèmes maître et esclave, ainsi que les résultats obtenus sur des instances de PDPTW de la littérature. Dans la section 4.2, nous intégrons à ce schéma trois critères de qualité de service sélectionnés pour leur intérêt applicatif et discutons des modifications à apporter à la méthode générale pour les prendre en compte. Ces trois critères sont la distance totale parcourue, le temps perdu en transport et le taux de remplissage des véhicules. L'optimisation suivant ces trois critères est évaluée sur des instances dérivées de celle de PDPTW de la littérature. Hormis l'intégration des critères de qualité de service, un des objectifs de ce chapitre est de calculer et de comparer les solutions obtenues en optimisant suivant différents critères. Afin que les résultats obtenus, en terme de qualité des tournées soient comparables et ne soient pas imputables à la méthode de résolution, une résolution exacte s'impose.

4.1 Méthode de résolution par génération de colonnes

4.1.1 Modélisation

Nous considérons le problème de DARP standard tel qu'il est défini dans la section 3.1. Trois types principaux de programmes linéaires en nombre entiers sont utilisés pour ce genre de problèmes : le modèle basé sur les arcs – ou modèle compact – servant de base aux méthodes de Séparation et Évaluation par Génération de Coupes (SEPG-COU), le modèle basé sur les tournées – ou modèle étendu – prenant la forme d'un problème d'affectation généralisé utilisé par les approches de type Séparation et Évaluation par Génération de Colonnes (SEPGCOL) et un modèle intermédiaire proposé

par [Mautor et Naudin \(2007\)](#) basé sur des états sur les arcs qui bien qu'aux caractéristiques prometteuses, offrant un compromis entre modèle étendu et compact en terme de relaxation linéaire et de nombre de variables, n'a encore suscité aucune application au DARP à notre connaissance.

Notre choix s'est porté sur le modèle étendu qui offre une décomposition intéressante du DARP. Comme nous l'avons remarqué dans la section 3.2, la plupart des contraintes de qualité de service sont évaluables au sein d'une même tournée. Ces contraintes qualifiées de locales, en faisant référence à la décomposition présentée en introduction à la génération de colonnes dans la section 2.2, sont toutes traitées dans le problème esclave sur lequel nous concentrons nos efforts. De plus, les problèmes de calcul de tournées de véhicules avec plusieurs véhicules, de par leur structure, sont généralement bien résolus grâce à ce modèle car le problème esclave est commun à tous les véhicules identiques.

On considère l'ensemble des tournées réalisables Ω . Une tournée ω est un plan de route complet attribuable à un véhicule. Les variables de décision binaires λ_ω sont telles que $\lambda_\omega = 1$ si la tournée est sélectionnée et $\lambda_\omega = 0$ sinon. L'indicateur binaire ρ_ω^r vaut 1 si la requête de transport r est prise en charge par la tournée ω et 0 sinon. À partir de ces données nous construisons le problème maître (MP), qui est un problème d'affectation généralisé. Par rapport au problème présenté dans la section 3.1, un seul type de véhicule est considéré. Considérer différents types de véhicules alourdit les notations sans pour autant amener des modifications majeures au niveau de la résolution, puisque cela générerait des problèmes esclaves similaires pour chaque type de véhicule. Avec R le nombre de requêtes et K le nombre de véhicules, on obtient la formulation suivante.

$$(MP) \quad \min \sum_{\omega \in \Omega} c_\omega \lambda_\omega \quad (4.1)$$

s.c.q.

$$\sum_{\omega \in \Omega} \rho_\omega^r \lambda_\omega = 1 \quad \forall r = 1, \dots, R, \quad (4.2)$$

$$\sum_{\omega \in \Omega} \lambda_\omega \leq K, \quad (4.3)$$

$$\lambda_\omega \in \{0, 1\} \quad \forall \omega \in \Omega. \quad (4.4)$$

Le coût, au sens général, d'une solution est la somme des coûts c_ω des tournées la composant. On impose par les contraintes (4.2), que chaque requête soit servie une fois exactement. La contrainte (4.3) limite le nombre de véhicules utilisés.

Nous nous proposons de calculer la solution optimale du programme linéaire obtenu par la relaxation des contraintes (4.4) qui donne les contraintes (4.5).

$$0 \leq \lambda_\omega \leq 1 \quad \forall \omega \in \Omega \quad (4.5)$$

L'ensemble Ω contenant un trop grand nombre de tournées, nous appliquons le principe de la génération de colonnes qui considère un sous-ensemble $\Omega' \subset \Omega$, donnant

ainsi un problème maître restreint (MPR). Une solution optimale de (MPR) est aussi solution optimale de (MP) s'il n'existe pas de variable λ_ω avec $\omega \in \Omega \setminus \Omega'$ de coût réduit négatif. La recherche d'une variable (colonne) de coût réduit négatif définit un problème esclave (EP) de type plus court chemin élémentaire avec contraintes de ressources (ESPPRC) que nous présentons ci-après. Nous reprenons les notations des données et variables de la définition du problème de la section 3.1. Les fenêtres de temps associées aux requêtes et évoquées dans la définition du problème générique de la section 3.1 sont construites comme suit. Les passagers de la requête r définissent une date de livraison au plus tard T_{r-}^{sup} . Un écart maximal entre cette dernière et la date de ramassage au plus tôt, noté T_r^{max} , est imposé. Les fenêtres de temps sont alors données par les relations (4.6). Nous supposons la durée de service constante y compris en un même arrêt où on somme les durées des services effectués. Ceci permet d'inclure les durées de service dans la durée des arcs et allège les notations. Nous rappelons que v_{R+1} et v_{-R-1} représentent respectivement les sommets des dépôts de départ et d'arrivée. Les véhicules peuvent se déplacer dans la fenêtre de temps $[0, T]$ et ont une capacité F .

$$\begin{aligned} T_{-R-1}^{inf} &= T_{R+1}^{inf} = 0, & T_{-R-1}^{sup} &= T_{R+1}^{sup} = T \\ T_{r+}^{inf} &= T_{r-}^{sup} - T_r^{max}, & T_{r+}^{sup} &= T_{r-}^{sup} - T_{r+r-} \\ T_{r-}^{inf} &= T_{r-}^{sup} - T_r^{max} + T_{r+r-} \end{aligned} \quad (4.6)$$

(EP) comporte des variables de décision binaires δ_{ij} telles que δ_{ij} est fixée à 1 si l'arc (v_i, v_j) appartient à la solution (tournée) et 0 sinon. Les variables t_i et f_i représentent respectivement la date de début du service i et le nombre de passagers dans le véhicule une fois le service i achevé.

Le coût de la tournée dans (EP) est le coût réduit de la tournée exprimé par l'équation (4.7), où π_ρ^r est la valeur de la variable duale associée à la contrainte (4.2) correspondante et π_κ celle de la contrainte (4.3). L'équation (4.8) définit ce coût en fonction de δ_{ij} dans le cas où c_ω est la distance parcourue et D_{ij} représente la longueur de (v_i, v_j) . Dans ce cas précis, le coût d'une tournée ne dépend pas des t_i et des f_i .

$$c_\omega^\pi = c_\omega - \sum_{r=1}^R \rho_\omega^r \pi_\rho^r - \pi_\kappa \quad (4.7)$$

$$f(\delta_{ij}, t_i, f_i) = \sum_{(v_i, v_j) \in \mathcal{A}} \delta_{ij} D_{ij} - \sum_{(v_r, v_j) \in \mathcal{A}} \pi_\rho^r \delta_{r+j} - \pi_\kappa \quad (4.8)$$

(EP)

$$\min c_\omega^\pi = f(\delta, t, f) \quad (4.9)$$

s.c.q.

$$\sum_{(v_i, v_j) \in \mathcal{A}} \delta_{ij} - \sum_{(v_j, v_i) \in \mathcal{A}} \delta_{ji} = 0 \quad \forall j \in \mathcal{R}^+ \cup \mathcal{R}^-, \quad (4.10)$$

$$\sum_{(v_i, v_j) \in \mathcal{A}} \delta_{ij} \leq 1 \quad \forall v_i \in \mathcal{V}, \quad (4.11)$$

$$\sum_{(v_{R+1}, v_j) \in \mathcal{A}} \delta_{R+1, j} = 1, \quad (4.12)$$

$$T_i^{inf} \leq t_i \leq T_i^{sup} \quad \forall v_i \in \mathcal{V}, \quad (4.13)$$

$$F_i^{inf} \leq f_i \leq F_i^{sup} \quad \forall v_i \in \mathcal{V}, \quad (4.14)$$

$$\delta_{ij}(t_i + T_{ij} - t_j) \leq 0 \quad \forall (v_i, v_j) \in \mathcal{A}, \quad (4.15)$$

$$\delta_{ij}(f_i + F_{ij} - f_j) = 0 \quad \forall (v_i, v_j) \in \mathcal{A}, \quad (4.16)$$

$$t \in \mathbb{R}^{|\mathcal{V}|}, \quad (4.17)$$

$$f \in \mathbb{N}^{|\mathcal{V}|}, \quad (4.18)$$

$$\delta \in \{0, 1\}^{|\mathcal{A}|}. \quad (4.19)$$

Les contraintes (4.10) assurent la continuité de la tournée (contraintes de conservation de flot), les contraintes (4.11) son élémentarité et la contrainte (4.12) son départ du dépôt. Le temps et les flux de passagers sont contraints par des fenêtres (contraintes (4.13) et (4.14)) et conservent leur flot par les contraintes (4.15) et (4.16). Les variables f_i sont déductibles des δ_{ij} mais sont introduites pour simplifier les formulations des contraintes. De même, les contraintes (4.15) et (4.16) ne sont pas exprimées sous une forme linéaire car, comme nous le verrons plus tard, le problème esclave est résolu par programmation dynamique et non par programmation linéaire en nombres entiers. Les fenêtres de flux $[F_i^{inf}, F_i^{sup}]$ sont définies par les équations (4.20) et 4.21).

$$F_i^{inf} = \begin{cases} F_i & \text{si } i \in \mathcal{R}^+ \\ 0 & \text{si } i \in \mathcal{R}^- \\ 0 & \text{si } i \in \{-R-1, R+1\} \end{cases} \quad (4.20)$$

$$F_i^{sup} = \begin{cases} F & \text{si } i \in \mathcal{R}^+ \\ F + F_i & \text{si } i \in \mathcal{R}^- \\ 0 & \text{si } i \in \{-R-1, R+1\} \end{cases} \quad (4.21)$$

Les critères de qualité de service supplémentaires sont intégrés au niveau du calcul du coût des tournées ou sous forme de contraintes.

4.1.2 Adaptation de la méthode générale de génération de colonnes

4.1.2.1 Résolution du problème maître restreint

Après avoir été mis sous la forme du programme linéaire en nombres entiers (4.1)-(4.4), nous résolvons le DARP dans un SEPGCOL où à chaque nœud le problème maître (MP) est résolu par une méthode de génération de colonnes tel que présenté dans la section 2.2. Les solutions primales et duales sont obtenues par un algorithme de programmation linéaire de type simplexe ou point intérieur, dont l'efficacité s'est montrée équivalente lors de nos expérimentations. Pour l'ensemble des expérimentations effectuées, le temps de résolution du programme linéaire se révèle négligeable par rapport à celui du problème esclave sur nos instances à quelques dizaines de requêtes. Nous ne gérons donc pas de manière particulière les colonnes générées par le problème esclave ; une colonne générée n'étant jamais supprimée de l'ensemble Ω' . La contrainte (4.3) sur la taille de la flotte peut être supprimée lorsque le nombre de véhicules n'est pas limitatif en pratique. Les distances étant toujours positives et respectant l'inégalité triangulaire, il est possible de relâcher la contrainte d'affectation (4.2) en une inégalité (≥ 1). Le problème maître devient alors un problème de couverture a priori plus simple. Les variables duales qu'il génère sont positives ou nulles, ce qui peut être un avantage lors de la résolution du problème esclave.

4.1.2.2 Représentation du problème esclave comme un ESPPRC

Toutes les contraintes de réalisabilité des tournées sont présentes dans le problème esclave. Nous modélisons le problème esclave comme un ESPPRC que nous résolvons par programmation dynamique. Sous l'effet des contraintes de fenêtres de temps assez serrées et des contraintes de précédence entre ramassage et livraison, la contrainte d'élémentarité des tournées a de fortes chances d'être peu active. Nous avons pourtant décidé de ne pas la relâcher car le contrôle des contraintes de couplage (et de précédence) entre ramassage et livraison reviennent aussi à contrôler l'absence de cycles relativement longs. Pour compenser le manque d'efficacité lié à la conservation des informations nécessaires au respect de la contrainte d'élémentarité, nous nous en servons pour renforcer les règles limitant l'extension des chemins partiels et la dominance entre chemins partiels. D'autre part, une partie des instances de PDPTW sur lesquelles nous avons effectué nos tests ont des fenêtres de temps très larges et le gain alors apporté par la résolution d'un ESPPRC est influent sur la qualité de la relaxation linéaire du problème maître. Nous détaillons la résolution du problème esclave dans la section 4.1.3.

4.1.2.3 Recherche arborescente

Pour les raisons discutées dans la section 2.2.2.1, la sélection des colonnes λ_ω comme variables de séparation est un mauvais choix. Le choix de variables de branchement le plus naturel est celui de l'arc « la plus fractionnaire » dans la solution optimale de la re-

laxation linéaire¹. La règle de branchement crée un sous-arbre où l'utilisation de cet arc est imposée et un sous-arbre où son utilisation est prohibée. Cette règle de branchement assure la convergence de la recherche. Elle peut être généralisée au flot sortant d'un ensemble de sommets (Naddef et Rinaldi, 2002). D'autres règles de branchement sont proposées dans la littérature, sur les véhicules (Desrochers et al., 1992), sur les fenêtres de temps (Dumas et al., 1991). Ropke (2005) teste quatre stratégies de branchements : sur les arcs, sur les véhicules puis les arcs, sur les flots sortants et sur les véhicules puis les flots sortants. Sur ces tests, le branchement sur les flots sortants se relève légèrement plus performant que celui sur les arcs. Brancher sur les véhicules suffit généralement à obtenir l'optimalité. Dans les cas où les véhicules ont un coût fixe d'utilisation très élevé, brancher sur les véhicules s'avère bien plus efficace car l'écart entre les bornes inférieures possibles a tendance à être très élevé.

La qualité de la relaxation linéaire, donc des bornes inférieures, du modèle d'affectation s'avère excellente, puisque sur l'ensemble des instances que nous avons résolu, presque toutes le sont à la racine. Par conséquent, nous nous contentons d'une seule règle de séparation assez simple. Nous sélectionnons la succession de services « la plus fractionnaire » dans la solution optimale de la relaxation linéaire ; ce qui revient à considérer le nombre de fois que sont utilisés tous les arcs réalisant une même succession de services. L'ordre de traitement des nœuds respecte la valeur de leur borne inférieure respective. Lorsque le nombre de véhicules est limité, la borne inférieure de chaque nœud peut être améliorée par la somme de la valeur optimale de la relaxation linéaire du problème maître et le produit entre la taille de la flotte et la valeur optimale du problème esclave, dans l'éventualité où cette dernière est connue. En effet, la valeur de la solution courante du problème maître restreint ne peut pas être plus diminuée que par l'intégration, pour chaque véhicule, d'une tournée correspondant à une solution optimale du problème esclave.

4.1.2.4 Initialisation de Ω'

L'ensemble des tournées initiales n'est pas forcément vide. Un algorithme de construction de tournées peut-être utilisé pour son initialisation. Cette initialisation nous affranchit de développements supplémentaires au niveau du problème esclave pour obtenir de bonnes solutions sur des problèmes peu contraints et propices aux heuristiques. Comme c'est le cas de certaines instances de Li et Lim (2001), où les services sont proches et où les fenêtres de temps et de flux d'occupation sont presque inexistantes. Pour ces problèmes, seule la preuve d'optimalité est difficile à obtenir.

À chaque nœud de l'arbre de recherche, lorsqu'on ne dispose pas de tournées formant une solution réalisable de (MPR), on utilise une tournée « passe-partout » – elle effectue tous les services – possédant un coût très élevé. On appelle cela la méthode du *BigM*. Si le problème est réalisable et qu'aucune colonne n'est générée par le problème esclave alors que la solution courante du problème maître restreint donne une

¹La fraction d'arc utilisée se mesure en sommant les valeurs des variables de sélection des tournées (λ_ω) incluant cet arc.

valeur non nulle à « passe-partout », on peut alors conclure que le coefficient attribué à « passe-partout » dans la fonction objectif est trop petit. Cependant, un coefficient trop élevé peut nuire aux performances de l’algorithme (Desaulniers, 2007). Lors de nos expérimentations, cet impact s’est révélé assez faible et l’utilisation de bornes supérieures même inatteignables (car trop élevées), n’a pas dégradé la rapidité de convergence de façon gênante, sachant que la limitation du temps de calcul ne fait pas partie de nos objectifs principaux.

La figure 4.1 résume le schéma général de notre implémentation de la méthode SEPGCOL. La borne inférieure associée à un nœud n est notée $BI(n)$ et BS désigne la meilleure borne supérieure courante. La valeur optimale du problème maître restreint est notée Z^* tandis que celle du problème esclave est notée c^* .

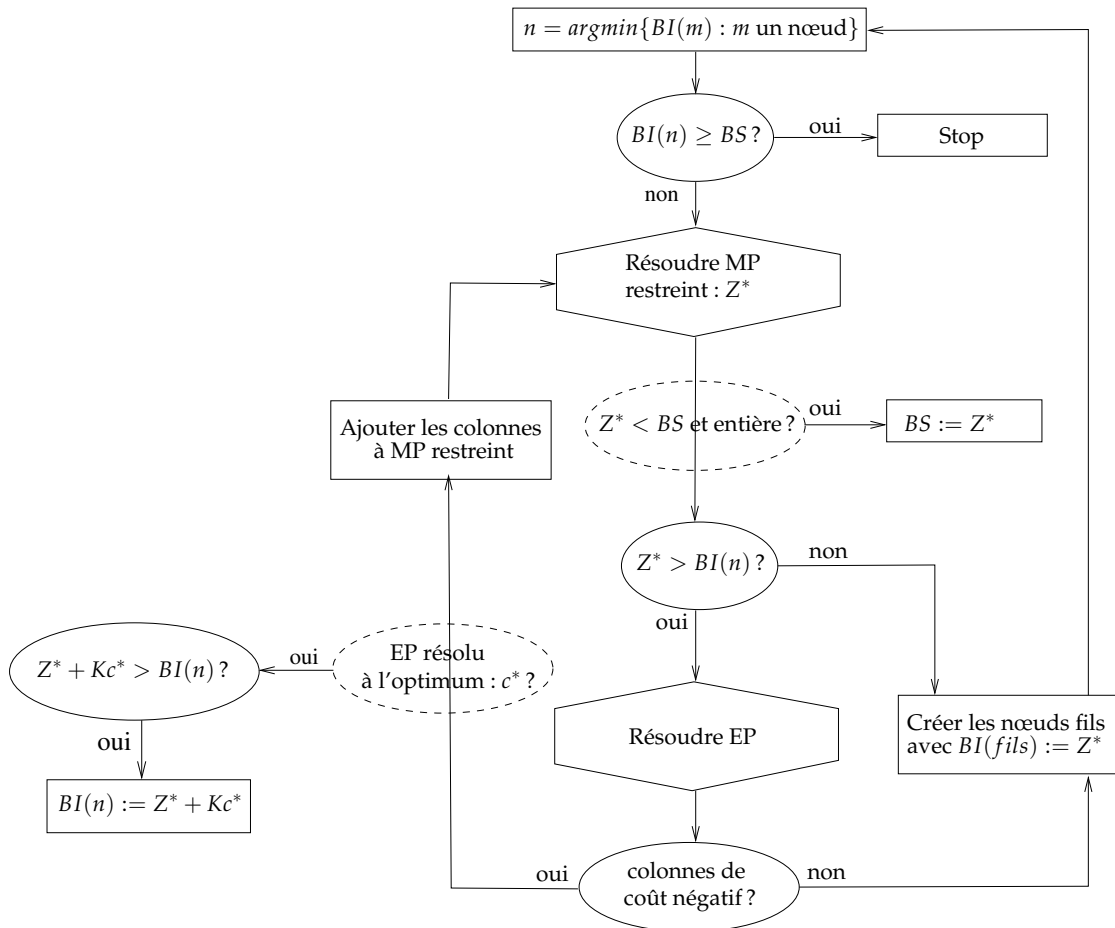


FIG. 4.1: Méthode de Séparation et Évaluation Progressive par Génération de Colonnes

4.1.3 Adaptation de la résolution du problème esclave

La résolution du problème est basée sur une simple adaptation au cas plus général de ESPPRC, de l'algorithme 1 de la section 2.3.2. Notre algorithme est donc défini par les règles d'extension et de dominance des chemins partiels issus du dépôt de départ. D'autres auteurs ont déjà utilisé cette approche (Sol, 1994; Sigurd et al., 2004b; Ropke, 2005).

Relativement à un chemin partiel ch , dont le sommet terminal est noté et_{ch} , nous définissons les trois ensembles de requêtes suivants : les requêtes en cours (ou ouvertes) par l'équation (4.22), les requêtes servies (ou fermées) encore atteignables par l'équation (4.23) et les requêtes inatteignables à cause de la ressource T par l'équation (4.24). La quantité de la ressource X ($X = F$ pour le flux de passagers, $X = T$ pour la durée de transport et $X = C$ pour le coût) consommée sur tout le chemin partiel ch est notée X_{ch} . On note $\mathcal{UFO}(ch) = \mathcal{U}(ch) \cup \mathcal{F}(ch) \cup \mathcal{O}(ch)$.

$$\mathcal{O}(ch) = \{r \in \mathcal{R} : r^+ \in ch, r^- \notin ch\} \quad (4.22)$$

$$\mathcal{F}(ch) = \{r \in \mathcal{R} : r^+, r^- \in ch, T_{ch} + T_{et_{ch}r^+} + T_{r^+r^-} \leq T_{r^-}^{sup}\} \quad (4.23)$$

$$\mathcal{U}(ch) = \{r \in \mathcal{R} : r^+ \notin ch, T_{ch} + T_{et_{ch}r^+} > T_{r^+}^{sup} \text{ ou } T_{ch} + T_{et_{ch}r^+} + T_{r^+r^-} > T_{r^-}^{sup}\} \quad (4.24)$$

4.1.3.1 Extension d'un chemin partiel

Pour chaque ressource X , nous définissons une fonction d'extension $ref_X(ch, (v_{et_{ch}}, v_j))$ d'un chemin partiel ch à travers un arc $(v_{et_{ch}}, v_j)$. D'après les équations (4.25) et (4.26), on constate aisément que chaque fonction d'extension est croissante². La formule (4.27) intègre la fonction d'extension du coût, exprimée par le coût de chaque arc et les valeurs duales issues du problème maître. Les valeurs initiales au dépôt de départ des trois fonctions d'extension sont nulles.

$$ref_T(ch, (v_{et_{ch}}, v_j)) = \max \left\{ T_{ch} + T_{et_{ch}j}, T_j^{inf} \right\} \quad (4.25)$$

$$ref_F(ch, (v_{et_{ch}}, v_j)) = F_{ch} + F_{et_{ch}j} \quad (4.26)$$

$$ref_C(ch, (v_i, v_j)) = \begin{cases} C_{ch} + D_{et_{ch}j} - \pi_\kappa & \text{si } j = -R - 1 \\ C_{ch} + D_{et_{ch}j} - \pi_\rho^j & \text{si } j \in \mathcal{R}^+ \\ C_{ch} + D_{et_{ch}j} & \text{sinon} \end{cases} \quad (4.27)$$

Lors de la phase d'extension nous contrôlons aussi l'ensemble des contraintes structurelles de précédence, de couplage et d'élémentarité. La contrainte de couplage paraît

²Nous considérons ici la croissance de la fonction par rapport aux niveaux de consommation de la ressource et non pas par rapport aux niveaux d'extension du chemin partiel.

essentielle car elle réduit très fortement le nombre de tournées réalisables. Le maintien pour chaque chemin partiel de l'ensemble des requêtes en cours permet de garantir précedence et couplage par simple parcours de cet ensemble. L'élémentarité nécessite de connaître aussi les requêtes finies encore atteignables (par rapport au temps). Certains auteurs, comme [Ropke \(2005\)](#), ne gèrent pas directement cette contrainte et résolvent un SPPRC. En effet, il est probable que peu de tournées non élémentaires soient générées. Le fait de conserver les requêtes en cours implique que la présence d'un cycle équivaut à la possibilité de servir deux fois entièrement une même requête. Ceci est improbable dans un problème aux fenêtres de temps relativement serrées. Cependant, un tel état de fait signifie aussi que l'ensemble des requêtes accomplies atteignables ($\mathcal{F}(ch)$) est constamment de petite taille pour une tournée partielle. De plus, cet ensemble permet de renforcer les critères de dominance, comme nous le montrons dans la section sur la dominance. Aussi, indépendamment de son utilisation pour le respect de la contrainte d'élémentarité, nous conservons l'ensemble des requêtes atteignables.

Une fois le sommet atteint par l'extension, une méthode idéale de validation de l'extension serait de résoudre le TSPTW (le problème du voyageur de commerce avec fenêtres de temps) passant par toutes les livraisons des requêtes en cours et finissant au dépôt. Ce problème est NP-difficile mais sa résolution reste envisageable pour des ensembles de requêtes en cours de très petite taille. [Dumas et al. \(1991\)](#) puis [Ropke \(2005\)](#) résolvent ce TSPTW sur des sous-ensembles de 3 requêtes avec un fort gain en temps de calcul, 50% en moyenne (ces comparatifs étant effectués sans l'utilisation d'autres améliorations de la méthode). Néanmoins, sans retenir cette approche, nous proposons des règles de validation plus souples basées sur une estimation (borne inférieure) sur la durée des arcs entrants minimaux aux livraisons des requêtes en cours, à partir d'une date donnée. Au départ d'un sommet i à la date t , nous notons cette estimation $entrantMin(i, t)$.

L'algorithme 3 réalise l'extension du chemin partiel $pere$ par l'arc (v_y, v_x) (où x est un service de la requête r). Le chemin partiel obtenu est noté ch . On a $et_{pere} = v_y$ et $et_{ch} = v_x$.

On commence par étendre les ressources (ligne 1) et contrôler le respect des fenêtres associées (lignes 2 – 5). La variable r_servie sert à identifier si la requête r est bien ouverte. Lorsque $x = r^-$, r_servie doit être à *vrai* (ligne 18). Ensuite (lignes 7 – 8), on initialise fin qui est une borne inférieure sur la date de fin des extensions de ch , et $maxTsup$ qui est une borne supérieure sur la plus grande date de livraison au plus tard des requêtes ouvertes de ch . Bien sûr, si fin est supérieur à $maxTsup$ (ligne 22) aucune extension de ch n'est valide ; ch peut donc être ignoré. Dans la boucle des lignes 9 à 17, on parcourt les requêtes ouvertes de $pere$ pour créer $\mathcal{O}(ch)$. On contrôle que la requête r ne soit pas ouverte si x est son ramassage (ligne 11), que les livraisons des requêtes ouvertes soient atteignables (ligne 14). On met à jour les variables fin , $maxTsup$ et r_servie . À la ligne 18, on vérifie que r soit ouverte si x est la livraison de r . Si atteindre la livraison r^- est possible, on peut débiter la requête r (lignes 19 à 21). Enfin, des lignes 23 à 26, on construit l'ensemble des requêtes fermées encore atteignables (par rapport au temps) de ch et on contrôle ne pas avoir déjà servi r (ligne 24).

Algorithme 3 : Algorithme d'extension d'un chemin partiel

Données : Un chemin partiel $pere$ et un arc d'extension (v_y, v_x) , x un service de la requête r .

Résultat : Le chemin partiel étendu ch ou un échec.

```

1  $F_{ch} := ref_F(pere, (v_y, v_x)); T_{ch} := ref_T(pere, (v_y, v_x)); C_{ch} := ref_C(pere, (v_y, v_x));$ 
2 si  $F_{ch} < 0$  alors échec;
3 si  $F_{ch} > F_x^{sup}$  alors échec;
4 si  $T_{ch} > T_x^{sup}$  alors échec;
5 si  $x = r^+$  et  $T_{ch} + T_{r^+r^-} > T_{r^-}^{sup}$  alors échec;
6  $r\_servie := faux;$ 
7  $fin := T_{ch};$ 
8  $maxTsup := T;$ 
9 pour chaque  $i \in \mathcal{O}(pere)$  faire
10   si  $i = r$  alors
11     si  $x = r^+$  alors échec;
12      $r\_servie := vrai;$ 
13   sinon
14     si  $T_{ch} + T_{xi^-} > T_{i^-}^{sup}$  alors échec;
15      $fin = fin + entrantMin(i^-, T_{ch});$ 
16      $maxTsup = \max \{ T_{i^-}^{sup}, maxTsup \};$ 
17      $\mathcal{O}(ch) \leftarrow i;$ 
18 si  $x = r^-$  et  $r\_servie = faux$  alors échec;
19 si  $x = r^+$  alors  $fin := fin + entrantMin(r^-, T_{ch});$ 
20  $maxTsup := \max \{ T_{r^-}^{sup}, maxTsup \};$ 
21  $\mathcal{O}(ch) \leftarrow r;$ 
22 si  $fin > maxTsup$  alors échec;
23 pour chaque  $i \in \mathcal{F}(pere)$  faire
24   si  $i = r$  alors échec;
25   si  $T_{ch} + T_{xi^+} + T_{i^+i^-} \leq T_{i^-}^{sup}$  alors  $\mathcal{F}(ch) \leftarrow i;$ 
26 si  $x = r^-$  et  $T_{ch} + T_{r^-r^+} + T_{r^+r^-} \leq T_{r^-}^{sup}$  alors  $\mathcal{F}(ch) \leftarrow i;$ 

```

Afin d'accélérer la phase d'extension en détectant les échecs le plus tôt possible, les ensembles \mathcal{O} et \mathcal{F} sont triés dans l'ordre croissant de leur date de ramassage au plus tard.

4.1.3.2 Dominance

Une règle de dominance doit permettre d'éliminer le chemin dominé sans perdre la possibilité d'obtenir une solution optimale. La relation de dominance est basée sur les deux règles suivantes. Un chemin réalisable – par rapport aux ressources – arrivant au dépôt d'arrivée domine tous les chemins arrivant au dépôt d'arrivée avec un coût supérieur. Et, un chemin ch domine un chemin ch' , si pour toute extension $ext(ch')$ de ch' , il existe une extension de ch dominant $ext(ch')$. La relation de dominance est antisymétrique et transitive. La propriété de transitivité donne tout son intérêt à l'utilisation de relations de dominance dans un algorithme de programmation dynamique, qui sans dominance, construit toutes les solutions réalisables. Au plus une règle de dominance est forte, au plus elle permet d'éliminer des solutions de la recherche. Il est donc important d'utiliser des règles de dominance fortes sans toutefois, être trop coûteuses en temps de calcul. À cause des contraintes de consommation de ressources (structurelles comprises), il est a priori difficile de déterminer s'il y a dominance entre deux chemins partiels quelconques.

Nous présentons, ici trois règles de dominance. Ces trois règles, R1, R2 et R3, s'établissent entre des chemins ayant la même extrémité terminale. La première règle R1 (équations (4.28 à 4.32)), considère le cas où ch' , par l'équation (4.30), ne peut pas être étendu en servant une requête déjà servie par ch ; soit parce que ch' l'a déjà servie, soit parce qu'il est trop tard. Les requêtes ouvertes étant les mêmes et ch arrivant plus tôt en et_{ch} , il est clair que ch domine ch' car toute extension de ch' est réalisable à partir de ch qui a un coût inférieur.

R1 :

$$et_{ch} = et_{ch'} \quad (4.28)$$

$$T_{ch} \leq T_{ch'} \quad (4.29)$$

$$\mathcal{F}(ch) \subseteq \mathcal{U}(ch') \cup \mathcal{F}(ch') \quad (4.30)$$

$$\mathcal{O}(ch) = \mathcal{O}(ch') \quad (4.31)$$

$$C_{ch} \leq C_{ch'} \quad (4.32)$$

Cette règle de dominance s'applique dès lors que les fonctions d'extension sont croissantes même lorsque les coûts ne respectent pas l'inégalité triangulaire. Nous remarquons que l'égalité entre les requêtes en cours implique l'égalité des flux de passagers, $F_{ch} = F_{ch'}$. L'inconvénient de R1 est qu'elle oblige ch et ch' à une grande similarité ($\mathcal{O}(ch) = \mathcal{O}(ch')$). Cet inconvénient est évité par la règle R2 définie par les règles (4.33)-(4.37).

R2 :

$$et_{ch} = et_{ch'} \quad (4.33)$$

$$T_{ch} \leq T_{ch'} \quad (4.34)$$

$$\mathcal{F}(ch) \subseteq \mathcal{UFO}(ch') \quad (4.35)$$

$$\mathcal{O}(ch) \subseteq \mathcal{O}(ch') \quad (4.36)$$

$$C_{ch} \leq C_{ch'} \quad (4.37)$$

À la différence de R1, R2 impose simplement par la règle (4.36) aux requêtes ouvertes pour ch de l'être aussi pour ch' . On en déduit l'égalité (4.38) sur les flux de passagers qui garantit que la contrainte de flux peut-être négligée lors de la recherche d'extensions de ch dominant celles de ch' .

$$F_{ch} + F_{r+\in\{\mathcal{O}(ch')\setminus\mathcal{O}(ch)\}} = F_{ch'} \quad (4.38)$$

La règle (4.35) assure que les requêtes servies par ch ne puissent pas commencées à être servies ultérieurement par une extension de ch' . Pour que R2 soit valide, il faut non seulement que les fonctions d'extension des ressources soient croissantes; mais aussi qu'elles soient à valeurs positives sur les arcs autorisés à ch' et interdits à ch ; et que l'inégalité triangulaire sur toutes les ressources (temps, flux de passagers et coût) soit respectée dans les triangles dont le sommet intermédiaire est une livraison. Or, en intégrant la valeur de la variable duale π_{rho}^r propre à une requête r , uniquement au niveau du coût des arcs sortants du ramassage, les conditions sont remplies si les coûts du problème original sont positifs et respectent l'inégalité triangulaire. La figure 4.2 illustre cette conservation partielle de l'inégalité triangulaire avec des coûts d'origine respectant l'inégalité triangulaire.

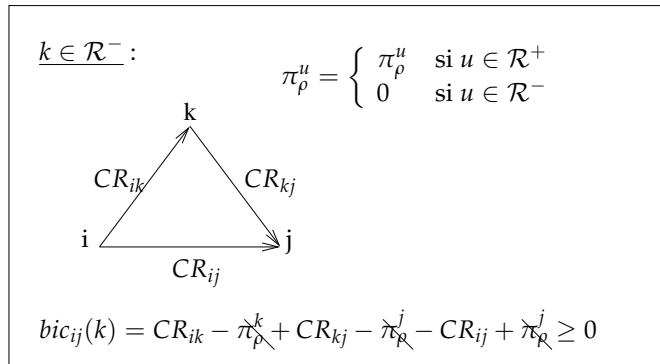


FIG. 4.2: Inégalité triangulaire conservée dans EP au niveau des livraisons

R2 est évidemment plus forte que R1 car toute dominance prouvée par R1 est aussi prouvée par R2. Un inconvénient de cette règle apparaît lorsque, suite par exemple à des branchements ou des réductions du graphe, l'utilisation d'arcs est forcée ou interdite. Le chemin ch peut se voir interdire l'accès à certains services accessibles pour des extensions de ch' . Il suffit pour cela qu'un service ne soit plus accessible, suite à des

réductions de graphe, que par des arcs issues de livraisons de requêtes de l'ensemble $\mathcal{O}(ch') \setminus \mathcal{O}(ch)$.

Enfin, nous définissons avec les règles (4.39)-(4.42), une dernière règle de dominance R3, où $BIC_2(E)$ est une borne inférieure sur le coût généré par des détours mettant en cause le ramassage et la livraison des requêtes de l'ensemble de requêtes E . $BIC_1(E)$ est le même type de borne restreint aux seules livraisons de E . Dans la règle (4.42), les deux bornes n'ont pas le même statut, BIC_1 prend en compte des détours par des livraisons obligatoires pour ch' vers lesquels ch ne peut pas être étendu sans effectuer le ramassage relatif. C'est pourquoi, cette borne est comptabilisée quelque soit son signe. À l'inverse, les extensions de ch' par des requêtes entières de $\mathcal{F}(ch)$ considérées dans $BIC_2(E)$, ne sont pas imposables à toute extension de ch' . La borne inférieure $BIC_2(E)$ est donc négative ou nulle.

R3 :

$$et_{ch} = et_{ch'} \quad (4.39)$$

$$T_{ch} \leq T_{ch'} \quad (4.40)$$

$$\mathcal{O}(ch) \subseteq \mathcal{O}(ch') \quad (4.41)$$

$$C_{ch} \leq C_{ch'} + BIC_1(\mathcal{O}(ch') \setminus \mathcal{O}(ch)) + BIC_2(\mathcal{F}(ch) \setminus \mathcal{UFO}(ch')) \quad (4.42)$$

L'intérêt de R3 par rapport à R2, est qu'elle fonctionne avec des coûts négatifs, ne respectant pas l'inégalité triangulaire. R3 reste valide malgré la suppression de certains arcs. Ces suppressions sont prises en compte dans la règle (4.42); éventuellement en fixant à $-\infty$ la borne inférieure pour les détours $((v_i, v_k), (v_k, v_j))$ des arcs (v_i, v_j) supprimés.

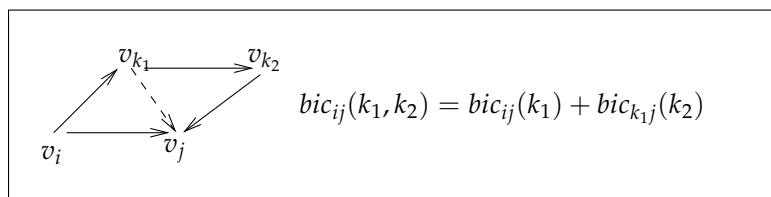
Nous proposons une méthode générique pour calculer ces bornes inférieures. Nous notons $bic_{ij}(k_1, k_2, \dots, k_n)$ le coût minimum engendré par un détour entre v_i et v_j passant par la suite de livraisons $(v_{k_1}, v_{k_2}, \dots, v_{k_n})$. Le coût $bic_{ij}(k_1, k_2, \dots, k_n)$ est donné par la formule (4.43) où CR_{ij} représente le coût de l'arc (v_i, v_j) dans le problème esclave. Il en découle l'égalité (4.44). La figure 4.3 illustre le cas d'un détour par deux livraisons k_1 et k_2 entre les services i et j .

$$bic_{ij}(k_1, k_2, \dots, k_n) = CR_{ik_1} + CR_{k_1k_2} + \dots + CR_{k_{n-1}k_n} + CR_{k_nj} - CR_{ij} \quad (4.43)$$

$$bic_{ij}(k_1, \dots, k_n) = bic_{ij}(k_1, \dots, k_{n-1}) + bic_{k_{n-1}j}(k_n) \quad (4.44)$$

Par définition, $bic(k_1, \dots, k_n)$ est le minimum des coûts pour tous les détours passant par $(v_{k_1}, v_{k_2}, \dots, v_{k_n})$ (équation (4.45)).

$$bic(k_1, \dots, k_n) = \min_{(v_i, v_j) \in \mathcal{A}} bic_{ij}(k_1, \dots, k_n) \quad (4.45)$$


 FIG. 4.3: Coût du détour par k_1 et k_2 de i vers j

On a donc l'inégalité (4.46).

$$\sum_{i=1}^n bic(k_i) \leq bic(k_1, k_2, \dots, k_n) \quad (4.46)$$

Il est alors facile de conclure que le coût minimal produit par les détours dus aux éléments de $\mathcal{O}(ch') \setminus \mathcal{O}(ch)$ est supérieur – donc moins bon pour une minimisation – à la somme des coûts minimaux de chaque livraison de $\mathcal{O}(ch') \setminus \mathcal{O}(ch)$, ce qui se traduit par l'inégalité (4.47)).

$$\sum_{r \in \mathcal{O}(ch') \setminus \mathcal{O}(ch)} bic(r^-) \leq BIC_1(\mathcal{O}(ch') \setminus \mathcal{O}(ch)) \quad (4.47)$$

De même, le coût associé aux requêtes de $\mathcal{F}(ch) \setminus \mathcal{UFO}(ch')$ est minoré par la somme des coûts négatifs de ces requêtes, comme dans l'inégalité (4.48).

$$\sum_{r \in \mathcal{F}(ch) \setminus \mathcal{UFO}(ch')} \min(0, bic(r^+) + bic(r^-)) \leq BIC_2(\mathcal{F}(ch) \setminus \mathcal{UFO}(ch')) \quad (4.48)$$

L'utilisation de règles de dominances « erronées », c'est-à-dire pouvant éliminer des chemins partiels dominants, permet de construire des heuristiques. Ce procédé est, à notre connaissance, peu répandu. Lorsque les tournées optimales contiennent beaucoup de requêtes, la programmation dynamique peut être assez longue à les trouver car elle explore beaucoup de tournées plus courtes avant de construire de longues tournées. Une règle de dominance privilégiant les tournées ayant effectué le plus de services est alors envisageable, par exemple en comparant les cardinaux des ensembles de requêtes fermées et ouvertes quitte à affaiblir les règles sur le temps et le coût. Pour notre part, nous avons préféré ne pas éliminer de chemins partiels éventuellement inclus dans une solution optimale du problème esclave, et avons uniquement agi sur l'ordre de traitement des chemins partiels (section 4.1.3.4) et en limitant temporairement l'extension de certains chemins partiels (section 4.1.3.5).

4.1.3.3 Réduction du graphe

En prétraitement, on peut réduire la taille du graphe en supprimant des arcs qui sont inutilisables.

Les arcs de G représentent des connexions entre paires de services. Pour des services de requêtes différentes, de telles connexions sont possibles si une des six configurations de la figure 4.4 entre les quatre services de deux requêtes mettant cet arc en jeu, est possible. Des fenêtres de temps serrées et des écarts importants entre ramassage et livraison réduisent fortement la densité du graphe. Ce prétraitement est fait une seule fois et reste valide pour toutes les itérations du problème esclave.

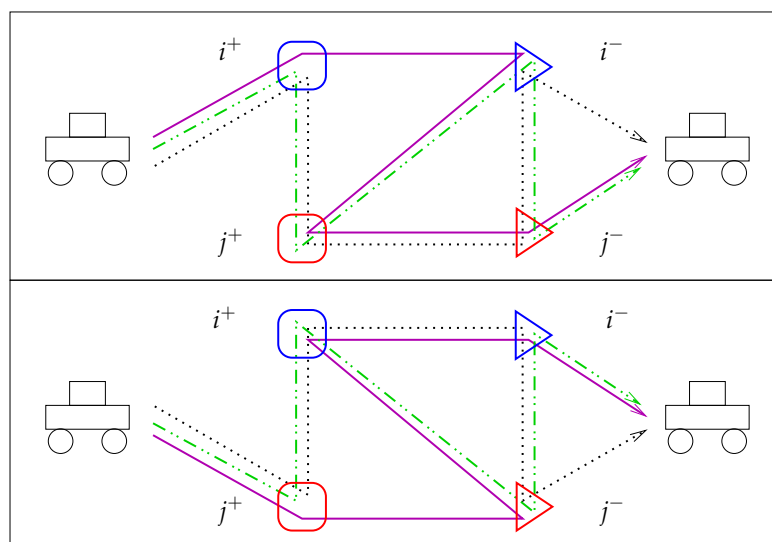


FIG. 4.4: Les six configurations pour servir deux requêtes

Une requête r peut être supprimée de (EP) en fonction de la valeur de sa variable duale associée π_{rho}^r qui peut rendre improbable sa présence dans une solution optimale de (EP).

La réduction du graphe peut être aussi utilisée comme heuristique par le retrait puis l'ajout progressif d'arcs pouvant appartenir à une solution optimale. De nombreuses stratégies sont proposées. Dumas et al. (1991) conservent un pourcentage des meilleurs arcs, Ropke (2005) limite le nombre de ramassages et de livraisons succédant à chaque service. Là-encore, nous pensons que la méthode de restriction temporaire de certaines extensions présentée dans la section 4.1.3.5, permet un comportement de l'algorithme de résolution équivalent.

4.1.3.4 Gestion des chemins partiels

L'ordre dans lequel sont traités les chemins partiels a une influence sur la rapidité avec laquelle l'algorithme va trouver des routes satisfaisantes. Les chemins partiels sont dans une liste de priorité. La priorité peut correspondre à un tri croissant suivant le temps consommé pour les chemins. Ce tri est adapté lorsque des solutions à coût réduit négatif sont difficiles à obtenir puisqu'il favorise les chemins partiels ayant le moins de chances d'être dominés par la suite ; d'où l'accroissement de la performance de la dominance. Cependant, afin d'améliorer les performances lorsque les seules bonnes routes

sont longues (en nombre de services) et assez nombreuses, nous trions les chemins partiels suivant leur coût réduit, favorisant ainsi les tournées prometteuses en terme de coût. Si après un temps de calcul paramétré à l'avance en fonction de la taille de l'instance, aucun chemin de coût négatif n'a été trouvé, nous utilisons un ordre basé sur le temps.

En fixant une limite au nombre de chemins partiels en attente de traitement, [Ropke \(2005\)](#) résolvent de manière heuristique le problème esclave basée sur la gestion des chemins partiels.

4.1.3.5 Recherche à limitation d'écart (LDS)

L'extension d'un chemin partiel vers tous les successeurs de son extrémité terminale provoque, malgré la dominance, une explosion rapide du nombre de chemins partiels traités. Nous proposons d'utiliser une méthode permettant de limiter le nombre de successeurs vers lesquels l'extension des chemins partiels est réalisée. Cette méthode dite de recherche à limitation d'écart (de l'anglais *Limited Discrepancy Search* ou LDS) proposée par [Harvey et Ginsberg \(1995\)](#), part du principe que lorsqu'une heuristique de branchement dans une recherche arborescente n'aboutit pas, le nombre de retours en arrière correspondant à de « mauvais » choix est assez faible. Ce procédé a été utilisé pour le VRPTW par [Rousseau et al. \(2004\)](#). Pour notre problème, cela revient à penser qu'une bonne tournée est composée de « bonnes » liaisons entre services voisins respectant des règles économiques et de seulement quelques liaisons ne respectant pas ces règles. Nous définissons donc un critère permettant de sélectionner des successeurs prometteurs à chaque service. Nous définissons aussi le nombre maximal de mauvaises liaisons que peut contenir une tournée. L'extension d'un chemin partiel se fait alors vers tous les « bons » successeurs et vers tous les autres si la limite de mauvaises liaisons n'est pas atteinte par ce chemin partiel. Nous appelons les chemins partiels n'ayant pas été étendus vers tous leurs successeurs, des « chemins partiels limités ». Si à la fin de l'algorithme aucune colonne de coût négatif n'est trouvée, il est relancé à partir des « chemins partiels limités » avec une limite d'écart augmentée. La recherche à limitation d'écart fournit ainsi une méthode exacte combinant à chaque itération les effets d'une réduction du graphe avec ceux d'une limitation du nombre de chemins partiels étudiés.

L'application du LDS à la recherche de chemins élémentaires de longueur 3 est illustrée par la figure 4.5 qui retrace les deux premières itérations de l'algorithme. Le poids indiqué sur les arcs représente le nombre d'écart cumulés pour arriver jusqu'à l'extrémité terminale du chemin finissant par cet arc. Toute extension atteignant la limite d'écart n'est pas effectuée. Lors de la première itération, cinq extensions (dont l'avant dernier sommet est entouré et dont l'arc terminal est en gras) sont « limitées » car elles atteignent la limite d'écart de 2. Ces extensions sont effectuées lors de l'itération suivante, et ce sont les extensions atteignant la limite d'écart de 3 qui sont alors « limitées ». Dans cet exemple, lorsqu'une extension vers un bon voisin est impossible – à cause de la contrainte d'éléментарité par exemple – on considère toujours la destination de cette extension comme un bon voisin. Les arcs marqués d'une croix représentent

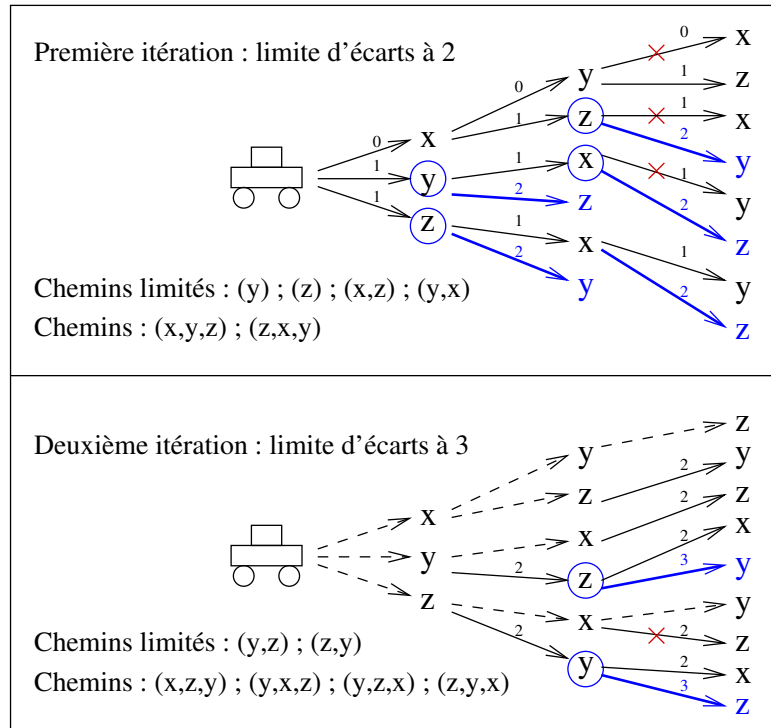


FIG. 4.5: Recherche de chemins élémentaires par un LDS paramétré à 1 bon voisin

des extensions violant la contrainte d'élémentarité.

Une des difficultés d'adaptation de cette méthode aux problèmes de type PDPTW est de trouver une cohérence entre les bons voisins du ramassage et ceux de la livraison d'une même requête. C'est pourquoi plutôt que de fixer un nombre de bons voisins, et afin de nous assurer que les liaisons correspondantes à une bonne configuration d'une paire de requête (voir la réduction du graphe section 4.1.3.3) soient considérées comme « bonnes » pour chaque service concerné, nous limitons les écarts uniquement vers des ramassages et réalisons systématiquement toutes les extensions vers les livraisons.

Le choix du critère de détermination des bons voisins est aussi important. Il peut être calculé en fonction d'une économie sur la fonction objectif du problème maître ou sur la consommation d'une ou de plusieurs ressources particulièrement contraignantes. Nous y incluons aussi les variables duales associées aux requêtes afin d'intégrer la fonction objectif du problème esclave d'une part et d'autre part, apporter un peu de diversification d'une itération à l'autre. Dans le cas de la minimisation de la distance, le critère d'évaluation des successeurs d'un ramassage i est donné par la formule (4.49) si on tient compte uniquement de la fonction objectif, et la formule (4.50) si on tient compte de la ressource temps.

$$\min_{j \in \mathcal{R}^+ \cup \mathcal{R}^-} \{D_{ij} - \pi_\rho^j\} \quad (4.49)$$

$$\min_{j \in \mathcal{R}^+ \cup \mathcal{R}^-} \{T_{ij} + T_j^{inf} - \pi_\rho^j\} \quad (4.50)$$

4.1.3.6 Résolution bidirectionnelle

Résoudre le SPPRC ou le ESPPRC, par programmation dynamique, en étendant simultanément des chemins partiels allant du dépôt de départ vers celui d'arrivée et d'autres chemins partiels réalisant le trajet inverse, réduit fortement le nombre de chemins partiels à considérer (Righini et Salani, 2006). En effet, la concaténation de deux chemins partiels de types différents permet de trouver des chemins assez longs à moindre coût en temps de calcul. Cependant, la présence de contraintes de couplage et de précedence entre les ramassages et les livraisons d'une même requête rend la concaténation de chemin moins performante car souvent génératrice de tournées invalides. Même si l'approche bidirectionnelle reste une piste d'amélioration à étudier dans le cas de problèmes de type *Ramassage & Livraison*, nous nous contentons d'une approche unidirectionnelle. Néanmoins, dans le soucis d'apporter un peu de diversification dans les colonnes remontées, nous résolvons le problème en partant alternativement du dépôt de départ et de celui d'arrivée. Il suffit d'adapter l'ensemble des règles d'extension et de dominance à un parcours inversé.

4.1.3.7 Utilisation des tournées connues

Les tournées présentes dans le problème maître peuvent servir de base à des heuristiques pour trouver de nouvelles tournées ayant un coût réduit négatif. Parmi les tournées du problème maître on s'intéresse aux tournées possédant les plus petits coûts réduits. On peut espérer qu'après quelques modifications leur coût réduit devienne négatif. Nous proposons un algorithme calculant la tournée de moindre coût dans un certain voisinage d'une tournée d'origine basé sur la méthode de programmation dynamique. L'idée est de conserver uniquement les requêtes ayant une valeur duale élevée dans l'ordre fourni par la tournée d'origine. Ensuite, si la séquence ainsi obtenue contient assez de requêtes (nous avons retenu la valeur 3 dans nos tests), en utilisant une adaptation de l'algorithme 1 présenté dans les paragraphes précédents, nous résolvons un problème identique au problème esclave auquel on ajoute la contrainte de respecter la séquence construite à partir d'une tournée existante. Cela se traduit par une contrainte supplémentaire sur l'extension des chemins partiels :

étant donné i et j deux services successifs dans la séquence, si i est le dernier service du chemin partiel appartenant à la séquence, alors toute extension vers un autre service que j doit pouvoir atteindre j à l'extension suivante.

La séquence renforce les contraintes de fenêtres de temps sur les services la composant (voir les équations (3.25) et (3.26)). Et, assurer l'extension vers le successeur du dernier service de la séquence présent dans un chemin partiel assure l'accessibilité à tous les autres services de la séquence dans le bon ordre. Les principales étapes de l'heuristique sont représentées dans la figure 4.6.

On représente la séquence par une ressource P dont la valeur est l'identifiant du prochain sommet à atteindre pour valider la séquence. La fonction d'extension de P d'un chemin partiel ch à travers l'arc (v_i, v_j) est donnée par la formule (4.51), où $successeur(x)$ est l'identifiant du sommet succédant à x dans la séquence, notée *Sequence*. La contrainte

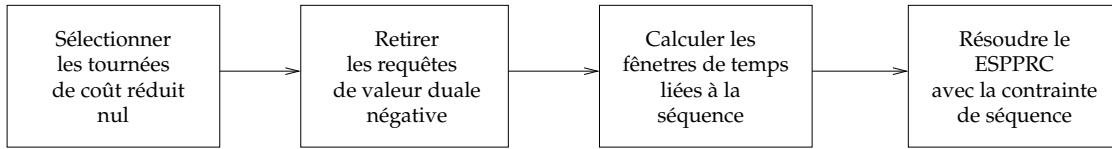


FIG. 4.6: Résolution heuristique du problème esclave

(4.52) est imposée sur cette ressource. Nous n’adaptions pas les règles de dominances et conservons les mêmes que pour la résolution complète.

$$ref_P(ch, (v_i, v_j)) = \begin{cases} successeur(P_{ch}) & \text{si } j = P_{ch} \\ P_{ch} & \text{si } j \notin Sequence \setminus \{P_{ch}\} \\ \text{échec} & \text{sinon} \end{cases} \quad (4.51)$$

$$T_{ch} + T_{jP_{ch}} \leq T_{P_{ch}}^{sup} \quad (4.52)$$

Dans notre implémentation, illustrée dans la figure 4.7, les requêtes conservées pour former la séquence sont celles dont la valeur duale associée est (strictement) positive. Les tournées du problème maître sélectionnées sont celles ayant le plus petit coût réduit, c’est-à-dire zéro. Parmi ces tournées, on trouve celles composant la solution courante (si elle existe) et celles représentant les autres variables de base. Après plusieurs itérations, les tournées de la solution, outre le fait d’avoir un coût réduit nul, peuvent être espérées intéressantes pour l’obtention de la solution optimale et les autres tournées « de base » pour l’obtention de la preuve d’optimalité.

D’autres méthodes permettent d’utiliser les tournées déjà calculées. De nombreuses heuristiques ou metaheuristiques semblent applicables comme des algorithmes de suppression/insertion et de recherche locale (Savelsbergh et Sol, 1998) ou de recherche à grand voisinage (Ropke, 2005). Ces deux heuristiques s’avèrent aussi très efficaces dans des versions de construction classique. Cependant, comme nous le constatons dans la section 4.2 et dans Garaix et al. (2007), l’intégration de certains critères de qualité de service originaux rend parfois le temps de calcul des opérateurs de voisinage classiques exponentiels et donc peu pertinents. Une autre méthode construit directement des chemins partiels à partir de tournées connues. En cours de résolution, par programmation dynamique par exemple, ces « bouts de tournée » (ou *metaextend* en anglais) peuvent être concaténés aux chemins partiels en construction. Rousseau et al. (2004) utilisent cette méthode dans le cadre de la résolution d’un VRPTW. Comme pour la résolution bidirectionnelle, les contraintes de couplage et de précédence risquent d’être un frein à l’efficacité de cette méthode.

4.1.4 Résultats sur des problèmes standard de la littérature

Il n’existe pas à l’heure actuelle de jeu d’instances référent pour le DARP. Ceci s’explique en partie par le lien étroit unissant ce problème à des applications pratiques, qui

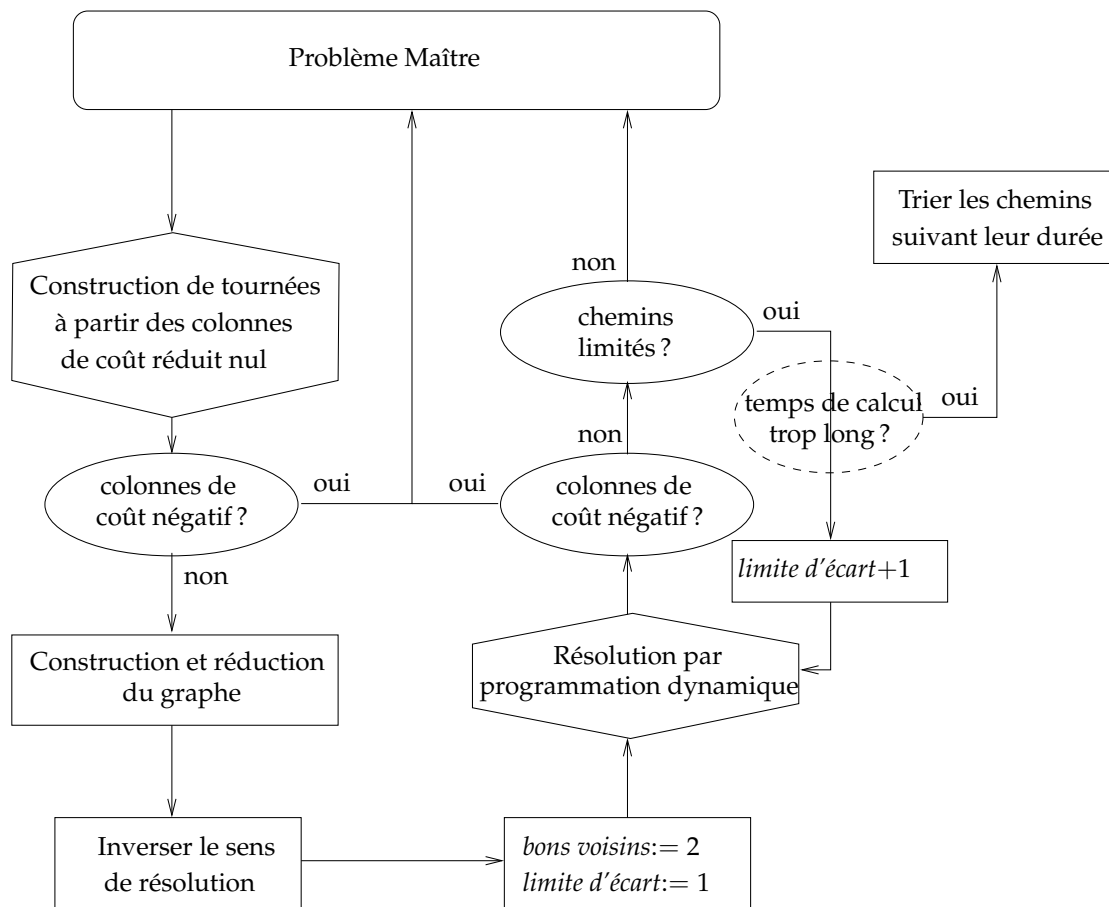


FIG. 4.7: Résolution du problème esclave par programmation dynamique

présentent des caractéristiques assez différentes pour produire des instances de DARP peu génériques. Cordeau (2006) met à disposition deux séries d'instances de DARP construites aléatoirement dans un repère euclidien de 20×20 . Malheureusement, elles correspondent à un DARP sortant du cadre que nous nous sommes fixés. Pour ces instances, la durée de transport des passagers doit respecter une borne prédéfinie, par contre les fenêtres de temps associées aux ramassages et aux livraisons sont de grande taille. Dans le DARP que nous considérons, la durée de transport est contrainte par des fenêtres de temps assez serrées pour la réalisation des services. De plus, la prise en compte de façon efficace dans notre méthode de la contrainte sur la durée de transport telle qu'elle est formulée dans les instances de Cordeau, n'est pas aisée. Cette dernière s'adapte en effet assez mal à une résolution du problème esclave par programmation dynamique.

Afin d'évaluer la performance de notre implémentation, nous l'avons testée sur la série 1 des instances de PDPTW de Li et Lim (2001) à 50 requêtes (environ), qui correspondent à un horizon de temps court entre 0 et 250 (sauf pour la classe c qui a un horizon de 1200). Ces instances sont construites à partir de solutions des instances à 100 requêtes de Solomon pour le VRPTW (Solomon, 1987). Ces instances sont elles aussi représentées dans un plan euclidien, et la distance en temps et en espace entre deux services est égale à la distance euclidienne séparant leurs deux sommets représentatifs. Elles sont divisées en trois classes, r pour *random* (aléatoires), c pour *cluster* (groupées) et rc un mélange des deux. Les ramassages des instances de Solomon ne pouvant pas être couplés avec un autre ramassage pour former une requête de type ramassage et livraison, sont transformées en des requêtes où ramassage et livraison sont confondus. La durée des services est de 10 pour r et rc et de 90 pour c . La proportion de services contraints par une fenêtre de temps est de 25%, 50%, 75% ou 100%. Les 25 véhicules sont identiques, basés au même dépôt et de capacité 200 ; ce qui permet en pratique de relâcher les contraintes sur le nombre de véhicules et sur la capacité.

La fonction objectif choisie est la minimisation de la distance totale parcourue, car elle permet de se comparer en terme d'efficacité à certains travaux sur le PDPTW et est un bon représentant des critères de qualité de service définissables par la somme du poids des arcs empruntés. Les résultats obtenus sont comparés avec la variante SP4 de la méthode de séparation et évaluation par génération de colonnes et coupes de Ropke (2005), qui est à notre connaissance la plus performante sur ces instances. Comme Ropke (2005), nous imposons une durée maximale de calcul de 2 heures. Les solutions primales et duales du problème maître sont obtenues par un solveur du commerce : CPLEX 9.0 en mode *Dual*, c'est-à-dire par l'algorithme du simplexe. L'ordinateur utilisé est un portable sous Linux cadencé à 800 Mhz et possédant 512 Mo de RAM. Les résultats obtenus apparaissent dans le tableau 4.1. Pour toutes les instances, une solution optimale, colonne *dist.*, est obtenue avec une preuve d'optimalité, sauf pour l'instance « lc104 » dont la preuve d'optimalité est obtenue après 2 heures. Le temps de calcul est donné dans la colonne *sec.* avec entre parenthèses le temps de calcul avec l'utilisation d'une heuristique d'initialisation dont la meilleure solution se trouve dans la colonne *heu.* $s(opt)$ donne le temps de calcul pour obtenir la solution optimale. La colonne *veh.* correspond au nombre de véhicules utilisés. $MP\%$ donne le pourcentage

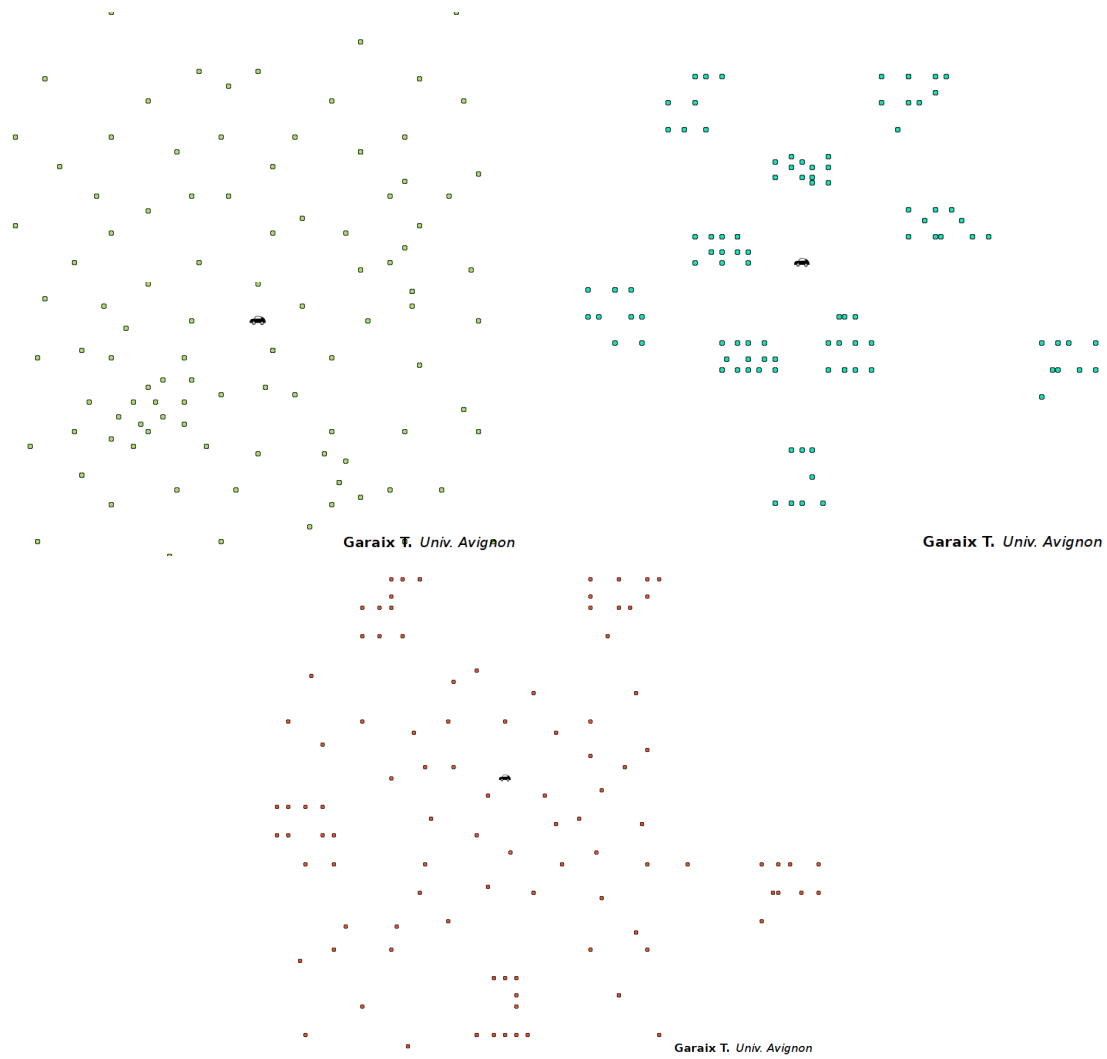


FIG. 4.8: Les 3 classes d'instances de Li et Lim

4.1. Méthode de résolution par génération de colonnes

nom	veh.	dist.	sec.	s(opt)	MP%	nœ.	heu%	heu s.	SP4 d.	SP4 s.
lr101	19	1650,8	0	0	0,0	1	1,9	0	1650,8	14
lr102	17	1487,6	4	2	0,0	1	9,7	6	1487,6	21
lr103	13	1292,7	39	8	0,0	1	6,8	28	1292,7	37
lr104	9	1013,4	X	871	0,0	1	11,1	314	1013,4	197
lr105	14	1377,1	1	1	0,0	1	6,0	1	1377,1	16
lr106	12	1252,6	4	2	0,0	1	8,4	3	1252,6	14
lr107	10	1111,3	149	37	0,0	1	12,0	21	1111,3	35
lr108	9	969,0	1293	161	0,0	1	15,5	1373	969,0	182
lr109	11	1209,0	7	5	0,0	1	5,9	6	1209,0	85
lr110	10	1159,4	3041	27	-0,1	5	9,9	2311	1135,9	X
lr111	10	1108,9	36	20	0,0	1	10,8	11	1108,9	253
lr112	9	1003,8	X	6065	0,0	1	18,6	2552	975,6	X
lc101	10	828,9	1	1	0,0	1	0,0	1	828,9	14
lc102	10	828,9	14	8	0,0	1	0,0	7	828,9	76
lc103	10	827,9	490	438	0,0	1	4,2	61	827,9	100
lc104	10	818,6	X	0	0,0	1	4,9	X	818,6	X
lc105	10	828,9	3	2	0,0	1	0,0	2	828,9	39
lc106	10	828,9	8	7	0,0	1	0,0	7	828,9	56
lc107	10	828,9	11	7	0,0	1	0,0	9	828,9	114
lc108	10	826,4	38	22	0,0	1	0,0	15	826,4	78
lc109	10	827,8	58	36	0,0	1	0,0	35	827,8	111
lrc101	15	1703,2	1	0	-0,1	5	2,3	1	1703,2	41
lrc102	12	1558,1	4	3	0,0	1	5,0	2	1558,1	22
lrc103	11	1258,7	68	56	0,0	1	14,6	10	1258,7	263
lrc104	10	1128,4	X	6857	0,0	1	11,4	139	1128,4	407
lrc105	13	1637,6	2	2	0,0	1	6,0	1	1637,6	23
lrc106	11	1424,7	2	2	0,0	1	10,8	2	1424,7	52
lrc107	11	1230,1	22	19	0,0	1	17,9	15	1230,1	90
lrc108	10	1147,4	68	44	0,0	1	7,3	46	1147,4	938

TAB. 4.1: Résultats sur les instances de Li et Lim à ≈ 50 requêtes

d'écart entre la valeur optimale de la relaxation linéaire du problème maître. Ce dernier est presque toujours nul sauf pour deux instances pour lesquelles on donne le nombre de nœuds créés (ils ne sont pas tous évalués) dans la recherche arborescente (colonne *na.*). Les deux dernières colonnes concernent la méthode SP4 de [Ropke et al. \(2007\)](#) et donnent la meilleure solution obtenue avec le temps de calcul, ou la borne inférieure (en italique) obtenue le cas échéant.

De l'observation des deux tableaux de résultats [4.1](#) et de caractéristiques [4.2](#), on constate que notre méthode est assez performante sur les instances possédant les contraintes les plus serrées. Pour les autres instances, l'heuristique de construction de tournées pallie un peu à l'inefficacité de notre algorithme. Mais quoiqu'il en soit, la résolution de ce type d'instances très peu contraintes ne nous intéresse pas directement. Dans tous les cas, la qualité de la borne fournie par la relaxation linéaire du problème maître est excellente et les temps de calcul sont comparables à ceux de SP4 pour les instances aux contraintes temporelles fortes.

À partir des données de la littérature, celles de Cordeau et celles de Li et Lim, nous avons dérivé de nouvelles instances aux contraintes de fenêtres de temps plus serrées et aux durées de service réajustées et égales au nombre de passagers ramassés (entre 1 et 6 pour les instances de Cordeau). Les tableaux [4.2](#) et [4.3](#) fournissent les caractéristiques de ces instances et de celles d'origine : le nom, le nombre de requêtes, la largeur moyenne des fenêtres de temps, l'écartement moyen entre les fenêtres de temps du ramassage et de la livraison d'une même requête ($T_{r^-}^{sup} - T_{r^+}^{inf}$), la densité du graphe et la quantité moyenne à transporter. La densité du graphe mesure le rapport entre le nombre d'arcs inter services (dépôts exclus) et le carré du nombre de ces services. Les arcs sont comptés après avoir appliqué les procédures de réduction du graphe de la section [4.1.3.3](#). Le nom des instances construites à partir de celles de Li et Lim débute par un 'g' à la place d'un 'l'. Le nom des instances construites à partir de celles de Cordeau est identique à l'original précédé d'un 'g' ; le deuxième nombre y indique le nombre de requêtes alors que le premier indique le nombre de véhicules disponibles pour le problème d'origine.

Pour les requêtes dont la date de ramassage au plus tôt est non nulle, les fenêtres de temps sont déduites à partir de cette date et d'un détour maximal représentant un tiers du plus court chemin en durée entre le ramassage et la livraison. Les fenêtres de temps des autres requêtes sont déduites à partir des dates de livraison au plus tôt, par un procédé similaire. Ces fenêtres de temps sont ensuite réajustées afin que toute requête puisse être servie au plus tôt, par un véhicule quittant le dépôt à la date 0. Afin de conserver la réalisabilité de ces instances, la contrainte sur le nombre de véhicules disponibles est relâchée. Les flux de passagers sont laissés à l'identique pour les instances de Cordeau, c'est-à-dire compris entre 1 et 6. Pour les instances de Li et Lim, le flux de passagers est égal à la partie entière supérieure du dixième du flux de la charge d'origine.

Les résultats des tableaux [4.4](#) et [4.5](#) illustrent l'efficacité accrue de notre méthode lorsque les contraintes de fenêtres de temps sont plus fortes. Ces résultats ont été obtenus avec un ensemble initial de tournées vide et le coefficient dans la fonction objectif de la tournée fictive « passe-partout » fixé à 3000.

4.1. Méthode de résolution par génération de colonnes

nom	req.	largeur		écart		densité		flux	
lr101	53	5,72	9,68	19,06	63,64	0,0485	0,0450	1,91	14,11
lr102	55	5,70	42,32	19,23	83,53	0,0687	0,2193	2,13	16,22
lr103	52	6,41	73,45	21,16	123,15	0,0735	0,3931	1,92	14,25
lr104	52	5,77	118,70	19,19	157,37	0,0659	0,6629	1,88	13,48
lr105	53	5,49	26,72	18,61	68,28	0,0524	0,0954	2,13	15,74
lr106	52	6,17	45,84	20,57	93,33	0,0602	0,2046	2,08	15,83
lr107	52	7,21	79,84	23,51	126,15	0,0715	0,4205	1,88	13,63
lr108	50	6,99	122,13	22,89	159,34	0,0642	0,7108	1,92	14,28
lr109	53	6,77	53,30	22,34	92,34	0,0460	0,2601	2,02	15,21
lr110	52	5,82	73,45	19,41	108,06	0,0453	0,4212	1,96	14,81
lr111	54	5,66	72,07	18,74	106,81	0,0532	0,4118	1,76	12,89
lr112	53	6,41	107,90	21,20	136,30	0,0239	0,7723	1,96	14,98
lc101	53	2,66	57,68	10,86	279,49	0,1120	0,1096	2,87	18,68
lc102	53	2,84	231,78	11,54	511,04	0,1139	0,3224	3,02	20,19
lc103	52	3,00	412,11	11,82	664,77	0,1133	0,5484	2,81	18,08
lc104	53	3,18	708,52	12,21	900,98	0,0873	0,8127	2,66	16,60
lc105	53	2,63	116,56	10,83	353,57	0,1076	0,1776	2,94	19,43
lc106	53	2,76	152,33	11,10	437,81	0,1054	0,2592	2,83	18,30
lc107	53	2,86	179,97	11,48	443,04	0,1038	0,2973	2,91	19,06
lc108	53	2,84	235,99	11,61	511,96	0,1005	0,4069	3,08	20,75
lc109	53	2,62	346,52	10,68	536,55	0,0976	0,5721	2,83	18,30
lrc101	53	6,18	27,99	20,95	73,30	0,0351	0,0724	2,40	16,62
lrc102	53	6,90	58,17	23,20	102,87	0,0571	0,2411	2,51	17,62
lrc103	53	5,51	88,32	18,97	125,19	0,0672	0,3930	2,43	17,15
lrc104	54	5,84	124,76	19,81	156,44	0,0551	0,6139	2,28	15,50
lrc105	54	6,74	45,32	22,68	85,76	0,0380	0,1591	2,44	17,09
lrc106	53	6,80	52,68	22,72	88,53	0,0395	0,1822	2,32	16,26
lrc107	53	6,05	76,29	20,65	111,36	0,0323	0,3580	2,49	17,72
lrc108	52	6,45	96,40	21,76	125,33	0,0245	0,5323	2,40	16,83

TAB. 4.2: Caractéristiques des instances de Li et Lim originales et modifiées (en gras)

nom	req.	largeur		écart		densité		flux	
gb2-16	16	5,48	369,45	16,45	734,88	0,1289	0,3496	3,00	3,00
gb2-20	20	4,72	376,20	14,17	746,85	0,1263	0,2094	4,20	4,20
gb2-24	24	5,65	369,26	16,96	734,83	0,1272	0,3056	3,63	3,63
gb3-18	18	5,34	369,41	16,02	734,50	0,1227	0,2863	3,78	3,78
gb3-24	24	4,74	375,49	14,23	745,46	0,1207	0,2782	3,63	3,63
gb3-30	30	5,70	366,19	17,09	728,77	0,1233	0,2883	3,57	3,57
gb3-36	36	5,34	382,87	16,01	761,42	0,1233	0,2660	3,69	3,69
gb4-16	16	6,04	381,08	18,12	759,25	0,1221	0,2021	4,38	4,38
gb4-24	24	4,55	382,60	13,64	759,29	0,1198	0,3116	3,58	3,58
gb4-32	32	5,08	389,88	15,24	774,91	0,1240	0,2729	3,81	3,81
gb4-40	40	5,21	364,10	15,64	723,63	0,1222	0,2266	3,95	3,95
gb4-48	48	4,76	353,09	14,28	700,71	0,1233	0,3064	3,25	3,25
gb5-40	40	5,66	380,11	16,99	756,55	0,1213	0,3447	3,20	3,20
gb5-50	50	5,54	363,17	16,63	722,42	0,1209	0,2824	3,64	3,64
gb5-60	60	5,44	394,09	16,33	784,07	0,1203	0,3899	3,15	3,15
gb6-48	48	5,48	372,33	16,45	740,63	0,1198	0,2632	3,81	3,81
gb6-60	60	5,50	371,50	16,49	739,00	0,1217	0,2892	3,62	3,62
gb6-72	72	5,20	367,11	15,61	729,63	0,1223	0,3020	3,49	3,49
gb7-56	56	5,46	383,64	16,39	763,21	0,1196	0,3100	3,57	3,57
gb7-70	70	4,65	369,35	13,94	732,99	0,1207	0,2770	3,80	3,80
gb7-84	84	5,69	376,03	17,06	748,44	0,1215	0,3299	3,36	3,36
gb8-64	64	4,82	377,94	14,46	750,52	0,1201	0,2947	3,61	3,61
gb8-80	80	5,01	378,71	15,03	752,44	0,1208	0,2775	3,80	3,80
gb8-96	96	5,01	362,89	15,04	720,80	0,1217	0,3355	3,30	3,30

TAB. 4.3: Caractéristiques des instances de Cordeau originales et modifiées (en gras)

4.1. Méthode de résolution par génération de colonnes

nom	veh.	dist.	sec.	MP%	nœud
gr101	24	1988,5	0	0,0	1
gr102	22	1927,2	1	0,0	1
gr103	17	1812,9	1	0,0	1
gr104	25	1897,6	0	0,0	1
gr105	25	2046,4	0	0,0	1
gr106	18	1864,0	0	0,0	1
gr107	17	1834,5	0	0,0	1
gr108	27	2125,7	0	0,0	1
gr109	23	1917,8	0	0,0	1
gr110	25	1961,4	0	0,0	1
gr111	21	1844,0	0	-0,6	3
gr112	27	2182,2	0	0,0	1
gc101	9	1002,8	5	0,0	1
gc102	9	1060,9	4	0,0	1
gc103	13	1250,4	3	0,0	1
gc104	28	1982,2	0	0,0	1
gc105	11	1185,4	3	0,0	1
gc106	13	1244,7	2	0,0	1
gc107	13	1305,9	3	0,0	1
gc108	12	1191,3	3	0,0	1
gc109	17	1453,2	2	0,0	1
grc101	22	2290,9	0	0,0	1
grc102	19	2145,7	0	0,0	1
grc103	17	2055,4	0	0,0	1
grc104	27	2599,8	0	0,0	1
grc105	25	2543,2	0	0,0	1
grc106	20	2179,9	0	0,0	1
grc107	29	2632,9	0	0,0	1
grc108	30	2662,9	0	0,0	1

TAB. 4.4: Résultats pour les instances de Li et Lim modifiées

nom	veh.	dist.	sec.	MP%	nœud
gb2-16	5	294,3	0	0,0	1
gb2-20	6	314,0	0	0,0	1
gb2-24	4	443,7	0	0,0	1
gb3-18	3	309,9	0	0,0	1
gb3-24	6	369,7	0	0,0	1
gb3-30	7	524,4	1	0,0	1
gb3-36	6	600,5	2	0,0	1
gb4-16	4	314,3	0	0,0	1
gb4-24	6	395,5	0	0,0	1
gb4-32	7	515,6	2	0,0	1
gb4-40	8	653,7	3	0,0	1
gb4-48	9	695,6	10	0,0	1
gb5-40	7	676,9	4	0,0	1
gb5-50	8	832,4	14	0,0	1
gb5-60	10	1000,2	29	0,0	1
gb6-48	8	773,5	10	0,0	1
gb6-60	9	935,1	30	0,0	1
gb6-72	10	1058,4	76	0,0	1
gb7-56	11	924,1	14	0,0	1
gb7-70	12	986,9	48	0,0	1
gb7-84	12	1323,2	167	0,0	1
gb8-64	11	948,3	38	0,0	1
gb8-80	10	1132,8	134	0,0	1
gb8-96	12	1355,7	437	0,0	1

TAB. 4.5: Résultats pour les instances 'b' de Cordeau modifiées

La méthode proposée se révèle assez performante sur des problèmes standard pour envisager de l'adapter à d'autres critères d'optimisation.

4.2 Intégration des critères de qualité de service

4.2.1 La distance totale parcourue

La distance totale parcourue est la somme des distances parcourues pour chaque tournée. Même si la distance parcourue d'une tournée ne se décompose pas par rapport aux uniques requêtes servies, elle est obtenue en sommant la distance parcourue entre chaque service, dépôts inclus. Ce critère rentre donc bien dans la troisième catégorie de notre classification, voir la section 3.2. Ce critère est représentatif de nombreux autres qui se calculent de façon similaire. Ce critère a un impact fort sur d'autres comme la durée totale de transport, le nombre véhicules utilisés et le coût d'exploitation. C'est la fonction objectif la plus souvent étudiée et c'est en l'optimisant que nous avons évalué notre algorithme. Dans cette section, la minimisation de la distance totale parcourue va nous permettre d'étudier l'intérêt de considérer des chemins alternatifs offerts par le réseau de transport. Ce thème est évoqué dans la section 3.3.3 où se trouve un schéma d'illustration de l'intérêt, en terme de coût de la solution, de considérer des chemins offrant des compromis entre la durée et la distance.

Nous supposons ici que la taille de la flotte est illimitée. On obtient donc le problème maître :

$$(MP) \quad \min \sum_{\omega \in \Omega} c_{\omega} \lambda_{\omega} \quad (4.53)$$

s.c.q.

$$\sum_{\omega \in \Omega} \rho_{\omega}^r \lambda_{\omega} = 1 \quad \forall r = 1, \dots, R, \quad [\pi_r^r] \quad (4.54)$$

$$\lambda_{\omega} \in \{0, 1\} \quad \forall \omega \in \Omega. \quad (4.55)$$

Le coût c_{ω} de chaque tournée est égal à la somme des distances des arcs la composant, comme indiqué dans l'équation 4.56.

$$c_{\omega} = \sum_{(v_i, v_j) \in \bar{\omega}} D_{ij} \quad (4.56)$$

4.2.1.1 Construction du p -graphe

Considérons le réseau routier composé de nœuds marquant l'intersection entre plusieurs tronçon. Nous construisons à partir de ce réseau, le 1-graphe $G_0 = (\mathcal{V}_0, \mathcal{A}_0)$, où \mathcal{V}_0 contient l'ensemble des nœuds du réseau routier plus un sommet pour chaque service. En un même arrêt, autant de sommets sont créés que de services y sont effectués.

À partir de chaque paire de sommets de \mathcal{V}_0 , on construit les arcs inclus dans \mathcal{A}_0 à partir des tronçons reliant les intersections ou arrêts correspondant à ces sommets.

À chaque arc trois poids sont attribués : la durée de parcours, la distance, et le flux de passagers³. L'algorithme 4 calcule l'ensemble des chemins Pareto-optimaux \mathcal{P}_{ij} entre chaque paire de services i et j . L'ensemble \mathcal{P}_{ij} correspond à l'ensemble des solutions Pareto-optimales du problème de réalisabilité associé au problème de plus court chemin avec contraintes de ressources (SPPRC) entre i et j , avec une ressource sur le durée, une sur le flux de passagers et une sur la distance. Ainsi, l'algorithme 4 est une simple adaptation de l'algorithme de programmation dynamique présenté dans la section 2.3.

Le tri des chemins partiels entre deux mêmes services par rapport à une ressource discriminante, comme la durée ou le coût, permet d'accélérer la phase d'insertion des nouveaux chemins partiels (des lignes 7 à 14).

Nous pouvons à présent considérer le p -graphe $G = (\mathcal{V}, \mathcal{A})$ où les éléments v_i de \mathcal{V} sont les services (départ et arrivée aux dépôts inclus) comme dans le cas d'un graphe simple, et l'arc $(v_i, v_j)^e$ ($e = 1, \dots, |\mathcal{P}_{ij}|$) représente un des chemins Pareto-optimaux de \mathcal{P}_{ij} .

4.2.1.2 Résolution du problème esclave

Les améliorations apportées à la résolution du problème esclave dans la section 4.1.3 sont aisément adaptables au cas du p -graphe. Les réductions de graphe nécessitent l'utilisation des arcs représentant les plus courts chemins en temps. Les valeurs duales associées aux services sont retranchées à l'ensemble des arcs entrants à ce service.

Considérer l'ensemble des itinéraires Pareto-optimaux offerts par le réseau routier augmente considérablement la taille du graphe à traiter même après les procédures de réduction évoquées dans la section 4.1.3.3. Nous proposons alors d'utiliser pour la résolution du problème esclave, un 1-graphe que nous appelons *graphe idéal* et notons G_{id} . De tous les arcs incidents aux deux mêmes sommets nous construisons un unique arc idéal qui prend la meilleure valeur pour chaque poids. Dans notre cas, l'arc idéal $(v_i, v_j)^{id}$ est défini par les équations (4.57)-(4.58). Pour la ressource d'occupation F , l'arc idéal prend la même valeur que tous les autres.

$$D_{ij}^{id} = \begin{cases} \min_{e=1, \dots, |\mathcal{P}_{ij}|} \{D_{ij}^e\} - \pi_{\rho}^i & \text{si } i \in \mathcal{R}^+ \\ \min_{e=1, \dots, |\mathcal{P}_{ij}|} \{D_{ij}^e\} & \text{sinon} \end{cases} \quad (4.57)$$

$$T_{ij}^{id} = \min_{e=1, \dots, |\mathcal{P}_{ij}|} \{T_{ij}^e\} \quad (4.58)$$

S'il n'y a pas de chemin de coût négatif dans G_{id} , alors il n'y en a sûrement pas dans G et le problème est résolu. À partir des chemins de coût négatif issus de G_{id} , nous

³Le flux de passager est nul pour les arcs sortants d'un nœud n'étant pas un service.

Algorithme 4 : Algorithme de construction du p -graphe

Données : G_0 : un graphe multivalué sur R ressources;
 \mathcal{P} : une liste de chemins partiels non traités;
 $\mathcal{P}_{e_i e_t}$: les listes de chemins partiels de e_i vers e_t ;
Résultat : $\mathcal{P}_{e_i e_t}$ où e_i et e_t sont des services

- 1 $\mathcal{P} \leftarrow$ un chemin partiel initial pour chaque service ;
- 2 **tant que** $\mathcal{P} \neq \emptyset$ **faire**
- 3 prendre $p \in \mathcal{P}$;
- 4 $e_t :=$ extrémité terminale de p ;
- 5 $e_i :=$ extrémité initiale de p ;
- 6 $domine :=$ faux ;
- 7 **si** e_t est un service **alors**
- 8 **pour chaque** chemin partiel $p' \in \mathcal{P}_{e_i e_t}$ **faire**
- 9 **si** $D_p \leq D_{p'}$ et $T_p \leq T_{p'}$ et $F_p \leq F_{p'}$ **alors**
- 10 $\mathcal{P}_{e_i e_t} := \mathcal{P}_{e_i e_t} \setminus \{p'\}$;
- 11 **sinon**
- 12 **si** $D_{p'} \leq D_p$ et $T_{p'} \leq T_p$ et $F_{p'} \leq F_p$ **alors**
- 13 $domine :=$ vrai ;
- 14 sortir boucle ;
- 15 **si** $domine =$ faux **alors**
- 16 **si** e_t est un service **alors** $\mathcal{P}_{e_i e_t} := \mathcal{P}_{e_i e_t} \cup \{p\}$ **pour chaque** arc (e_t, i) **faire**
- 17 $p' :=$ l'extension de p par (e_t, i) ;
- 18 **si** i est un service **alors**
- 19 **si** $T_{p'} \leq T_i^{sup}$ et $F_{p'} \leq F_i^{sup}$ **alors**
- 20 $\mathcal{P} := \mathcal{P} \cup \{p'\}$;
- 21 **sinon**
- 22 $\mathcal{P} := \mathcal{P} \cup \{p'\}$;

cherchons des chemins de coût négatif dans G . Pour cela nous cherchons à trouver une sélection d'arcs valide permettant de conserver la séquence du chemin trouvé intacte et minimisant le coût. Ce problème est présenté sous le nom de problème de sélection d'arcs dans une séquence fixée (FSASP) dans la section 3.3.3 traitant du problème d'horodatage dans un p -graphe linéaire. Un algorithme de résolution y est proposé (l'algorithme 2). Si la recherche dans les séquences obtenues à partir des chemins du graphe idéal est infructueuse, nous résolvons alors le ESPPRC directement sur G . Là encore l'adaptation de l'algorithme 1 est aisée, puisqu'il suffit lors de la phase d'extension d'étendre les chemins partiels par tous les arcs sortants de son extrémité terminale.

Dans la figure 4.9, nous montrons l'intégration de ces étapes dans le schéma général de résolution. Les éléments nouveaux apparaissent en gras dans des cadres à bords épais.

4.2.2 Le temps perdu

Hormis le temps de transport passé dans le véhicule, le passager est parfois obligé d'attendre le véhicule ou d'attendre à son arrivée, attentes qui lui sont imposées par le transporteur. Le premier cas, appelé requête *retour*, peut correspondre à une personne souhaitant rentrer chez elle en sortant du travail à une heure précise et le second cas, la requête *aller*, à une personne se rendant à un rendez-vous et n'ayant aucun intérêt à être en avance. Ces deux types de requêtes forment les ensembles disjoints *Aller* et *Retour*. Le premier type de passager souhaite donc être livré le plus tôt possible et le second type être ramassé le plus tard possible. Nous appelons *temps perdu* l'écart entre la date de livraison effective et la date de livraison au plus tôt pour les requêtes *retour* (équation (4.59)) et l'écart entre la date de ramassage effective et la date de ramassage au plus tard pour les requêtes *aller* (équation (4.60)). Ce critère de qualité de service est présenté plus en détail dans la section 3.2 dans le paragraphe réservé aux écarts par rapport à des dates de service souhaitées.

$$\text{tempsperdu}(r) = t_{r-} - T_{r-}^{\text{inf}}, \forall r \in \text{Retour} \quad (4.59)$$

$$\text{tempsperdu}(r) = T_{r+}^{\text{sup}} - t_{r+}, \forall r \in \text{Aller} \quad (4.60)$$

Le temps perdu se mesure indépendamment pour chaque requête. Nous évaluons, au niveau d'une tournée puis d'une solution, le niveau de qualité de service sur ce critère en sommant le temps perdu pour chaque requête. L'objectif est alors de minimiser cette somme. Nous remarquons que pris comme seul critère d'optimisation, si on dispose d'au moins autant de véhicules que de requêtes, une solution optimale triviale est de servir chaque requête par un véhicule différent. Dans un TAD traditionnel, ce critère a du sens soit combiné à un autre ayant un effet minimisant sur le nombre de véhicules, soit accompagné d'une contrainte forte sur la taille de la flotte disponible. La contrainte d'affectation peut être relâchée en une inégalité puisqu'il est trivial de construire une solution équivalente à partir d'une solution optimale servant plusieurs fois la même requête.

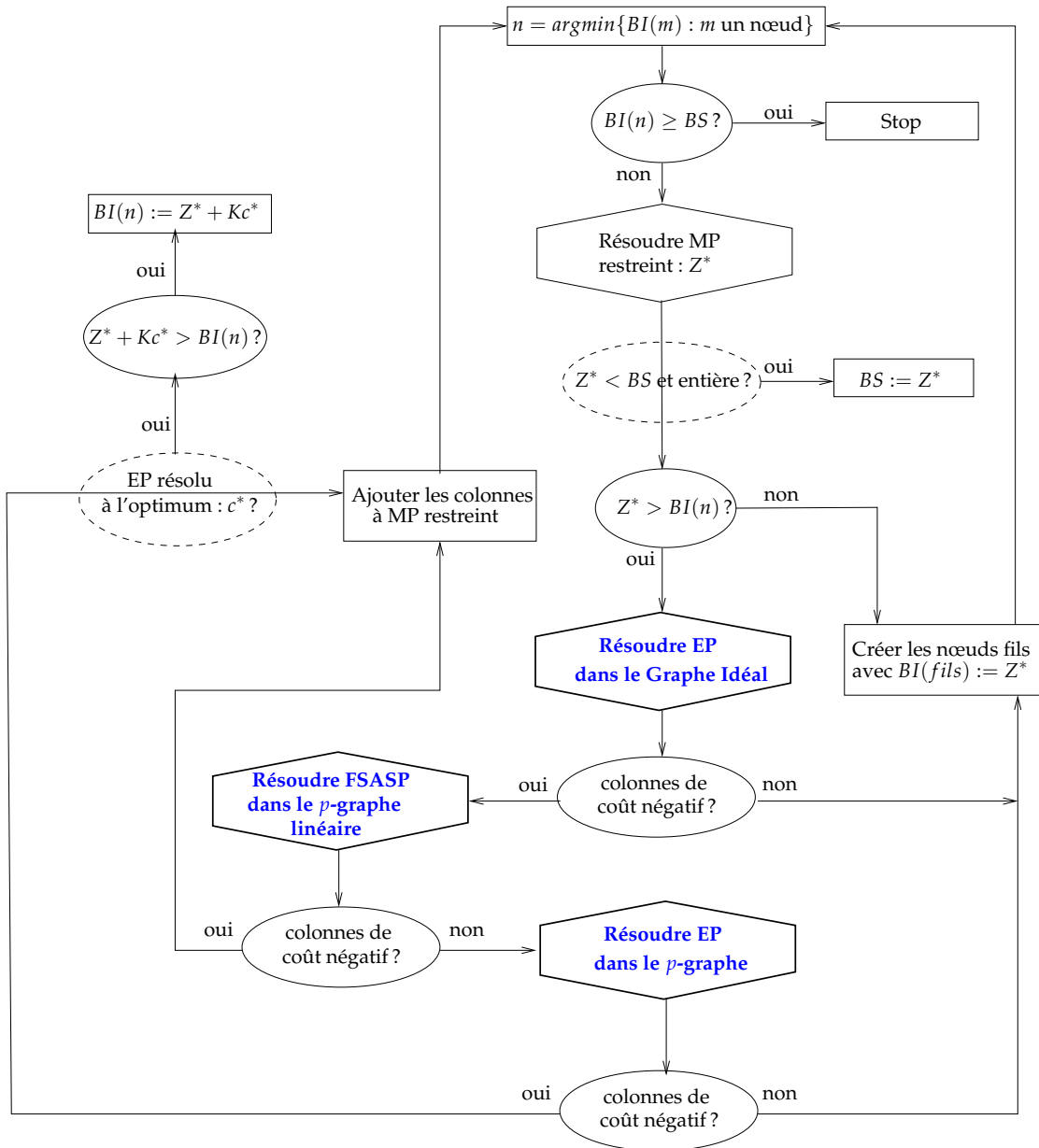


FIG. 4.9: Adaptation de la méthode à la distance totale parcourue

Le problème maître MP prend donc la forme suivante :

$$(MP) \quad \min \sum_{\omega \in \Omega} c_{\omega} \lambda_{\omega} \quad (4.61)$$

s.c.q.

$$\sum_{\omega \in \Omega} \rho_{\omega}^r \lambda_{\omega} \geq 1 \quad \forall r = 1, \dots, R, \quad [\pi_{\rho}^r] \quad (4.62)$$

$$\sum_{\omega \in \Omega} \lambda_{\omega} \leq K \quad [\pi_{\kappa}] \quad (4.63)$$

$$\lambda_{\omega} \in \{0, 1\} \quad \forall \omega \in \Omega. \quad (4.64)$$

Le coût c_{ω} de chaque tournée est égal à la somme des temps perdus pour chaque requête ; comme indiqué dans l'équation (4.65) dans le cas des requêtes *aller*.

$$c_{\omega} = \sum_{r^+ \in \dot{\omega}} T_{r^+}^{sup} - t_{r^+} \quad (4.65)$$

4.2.2.1 Résolution du problème esclave

4.2.2.1.1 Cas de requêtes retour uniquement Rien ne change par rapport au modèle standard proposé dans la section 4.1.1 si ce n'est la ressource liée à la fonction objectif. Le coût n'est pas défini directement sur les arcs, il se calcule dynamiquement par sa fonction d'extension décrite par l'équation 4.66. En effet, les dates de service (t_i) obtenues par la fonction d'extension définie par la formule 4.25 ne sont pas remises en cause par les extensions futures, ni pour des raisons de réalisabilité – les fenêtres de temps étant entièrement compatibles – ni pour des raisons d'optimalité – les dates de ramassage au plus tôt étant optimales pour ce critère. Les règles de dominance entre chemins partiels restent les mêmes. En effet, les durées respectant l'inégalité triangulaire, tout détour par une livraison entraîne une augmentation du coût. De plus, le fait d'arriver plus tôt au sommet terminal pour le chemin partiel dominant assure que ses extensions peuvent arriver au moins aussi tôt, que celles du chemin partiel dominé, pour les livraisons des requêtes en cours. À ce titre, les règles de dominance sur l'inclusion des requêtes en cours et les coûts sont valides.

$$ref_C(ch, (v_i, v_j)) = \begin{cases} C_{ch} & \text{si } j \in \mathcal{R}^+ \\ C_{ch} + t_j - T_j^{inf} - \pi_{\rho}^j & \text{si } j \in \mathcal{R}^- \\ C_{ch} - \pi_v^v & \text{si } j = -R - 1 \end{cases} \quad (4.66)$$

Il est notable que cette procédure n'est pas symétrique et que partir du dépôt d'arrivée provoque un affaiblissement de la dominance puisque les dates optimales des services présents dans un chemin partiel « inversé » ne peuvent pas être déterminées avec exactitude dans ce cas.

4.2.2.1.2 Cas de requêtes aller uniquement Dans ce cas, ce sont les dates de ramassage au plus tard qui sont optimales. Une procédure parfaitement symétrique à celle proposée pour les requêtes *retour*, construisant les tournées du dépôt d'arrivée vers celui de départ permet de résoudre le problème.

4.2.2.1.3 Cas de requêtes aller et retour Dans ce cas, il est impossible de déterminer les dates optimales de service lorsque la tournée est en construction. Il est donc nécessaire d'affaiblir les règles de dominance liées aux dates de service. Cependant, l'incertitude sur une dominance ne porte que sur le coût final des tournées et non sur leur réalisabilité. Il suffit donc de calculer une borne supérieure sur le coût de la tournée partielle pour pouvoir effectuer la comparaison. Une borne supérieure sur le coût est obtenue par résolution du programme linéaire BS_{ch} où le graphe $G_{ch} = (\mathcal{V}_{ch}, \mathcal{A}_{ch})$ représente le chemin partiel ch et $Aller(ch)$ et $Retour(ch)$ les ensembles de requêtes *aller* et *retour* servies dans ch . Pour i et j deux services successifs dans ch , δ_{ij} est fixé à 1 et à 0 pour tout autre paire de services.

(BS_{ch})

$$\max \sum_{r \in Retour(ch)} (t_{r^-} - T_{r^-}^{inf} - \pi_{\rho}^r) + \sum_{r \in Aller(ch) \cup (\mathcal{O}(ch) \cap Aller)} (T_{r^+}^{sup} - t_{r^+} - \pi_{\rho}^r) \quad (4.67)$$

s.c.q.

$$T_i^{inf} \leq t_i \leq T_i^{sup} \quad \forall v_i \in \mathcal{V}_{ch}, \quad (4.68)$$

$$t_i + T_{ij} - t_j \leq 0 \quad \forall (v_i, v_j) \in \{\mathcal{A}_{ch} : \delta_{ij} = 1\}, \quad (4.69)$$

$$t_i \geq 0 \quad \forall v_i \in \mathcal{V}_{ch}. \quad (4.70)$$

On note $z^*(BS_{ch})$ la valeur optimale de BS_{ch} et $z^*(BI_{ch})$ la valeur optimale de BI_{ch} qui est le pendant de BS_{ch} pour la minimisation. Ainsi $z^*(BI_{ch})$ est une borne inférieure sur le coût de ch . On remplace la règle de dominance sur les coûts ($C_{ch} \leq C_{ch'}$) entre ch et ch' par la règle 4.71.

$$z^*(BS_{ch}) \leq z^*(BI_{ch'}) \quad (4.71)$$

Cette règle est valide puisque pour toute requête *retour* en cours, le fait que T_{ch} soit inférieur ou égal à $T_{ch'}$ assure la dominance en terme de coût pour les livraisons des requêtes *retour* en cours. Pour les autres requêtes servies ou commencées, la règle compare le pire cas pour ch et le meilleur pour ch' .

Pour calculer le coût de la tournée complète, il suffit d'utiliser le même modèle en remplaçant la maximisation par une minimisation et en retranchant la variable duale associée au véhicule.

4.2.3 Le taux de remplissage

Un critère permettant d'évaluer la pertinence d'un TAD et son éventuelle saturation, est le taux de remplissage des véhicules roulants. Cet indicateur de performance peut aussi être pris comme critère d'optimisation puisqu'il garantit une certaine efficacité du TAD et intervient comme outil de négociation entre transporteurs et collectivités locales qui sont prêtes à subir la circulation de véhicules (coût financier, congestion du trafic routier, pollution...) à condition qu'ils soient bien remplis. Cependant, un tel modèle doit raisonnablement être contraint pour ne pas défavoriser les utilisateurs (chauffeurs et passagers). Il paraît, en effet, inconcevable de « promener » des passagers uniquement pour qu'ils remplissent bien le véhicule et qu'ainsi ils fassent augmenter le taux de remplissage global des véhicules du système. À cet effet, nous imposons que pour une séquence de service fixée, l'horodatage soit fait en minimisant le temps perdu. Cela garantit une qualité de service « visible » correcte pour les passagers qui n'ont pas l'impression de faire des détours aberrants. La flotte disponible se limite généralement à quelques types de véhicules de capacité fixe. Afin d'estimer le niveau d'utilisation du service il est préférable d'évaluer, donc d'optimiser, le taux d'inoccupation des véhicules plutôt que leur taux de remplissage, car le taux d'inoccupation tient mieux compte de la capacité des véhicules (voir la section 3.2). La méthode proposée s'adapte facilement à ces deux critères. Pour garder des notations allégées, nous présentons ici le taux de remplissage. Le problème maître prend une forme fractionnaire avec D_ω représentant la durée totale roulante de la tournée ω (équation (4.72)) et N_ω représentant son remplissage ; c'est-à-dire la somme des produits entre la durée et le nombre de places occupées pour chaque tronçon roulant de la tournée (équation (4.73)). L'horodatage des tournées suivant le temps perdu n'a donc pas d'impact sur la fonction objectif.

$$D_\omega = \sum_{(v_i, v_j) \in \tilde{\omega}} T_{ij} \quad (4.72)$$

$$N_\omega = \sum_{(v_i, v_j) \in \tilde{\omega}} f_i T_{ij} \quad (4.73)$$

Dans le cas où on désire mesurer N_ω et D_ω sur l'ensemble de la tournée et non uniquement sur les parties roulantes, imposer des contraintes de type temps perdu a un impact sur la fonction objectif. Cependant, il reste raisonnable de considérer que pour une séquence fixée, la fonction objectif calculant les heures de passage soit différente de celle du problème maître, car un passager mesure la qualité de l'horodatage de la tournée uniquement dans le cadre de la séquence d'arrêts qui lui est proposée.

Si on suppose le nombre de véhicules disponibles non contraignant, on peut supprimer la contrainte sur les véhicules. En revanche, la relaxation de la contrainte d'affectation en une inégalité est incompatible avec la maximisation du taux de remplissage.

Le problème maître restreint (MPR) est donc le suivant :

(MPR)

$$\max \frac{\sum_{\omega \in \Omega} N_\omega \lambda_\omega}{\sum_{\omega \in \Omega} D_\omega \lambda_\omega} \quad (4.74)$$

s.c.q.

$$\sum_{\omega \in \Omega} \rho_{\omega}^r \lambda_{\omega} = 1 \quad \forall r = 1, \dots, R, \quad [\pi_{\rho}^r] \quad (4.75)$$

$$\lambda_{\omega} \geq 0 \quad \forall \omega \in \Omega. \quad (4.76)$$

L'optimisation de programmes fractionnaires (aux contraintes linéaires) est un domaine étudié de longue date. [Isbell et Marlow \(1956\)](#) présentent déjà un premier algorithme inspiré du simplexe. Pour résoudre notre problème fractionnaire linéaire, nous proposons deux méthodes de résolutions, une méthode itérative basée sur l'algorithme de [Dinkelbach \(1967\)](#) puis une méthode directe ajoutant une contrainte au problème maître et basée sur la transformation proposée par [Charnes et Cooper \(1962\)](#).

4.2.3.1 La méthode itérative

4.2.3.1.1 Résolution du problème maître Pour résoudre le programme linéaire fractionnaire (MPR), nous utilisons l'algorithme itératif de [Dinkelbach \(1967\)](#) qui permet de traiter des fonctions objectifs au numérateur concave et au dénominateur convexe. Soit le rapport N/D ($D \neq 0$, $N = \sum_{\omega \in \Omega} N_{\omega} \lambda_{\omega}$ et $D = \sum_{\omega \in \Omega} D_{\omega} \lambda_{\omega}$) à maximiser, on construit une suite de problèmes de la forme $N - q_n D$. Le principe de l'algorithme est de construire une suite de rapports (q_n) convergente vers l'optimum du rapport N/D . L'algorithme de Dinkelbach est décrit par la figure 4.10. Cet algorithme est convergent vers une solution dont on fixe au préalable le coefficient d'approximation $\epsilon > 0$.

Après avoir remarqué que N et D peuvent être supposés non nuls, appliquer cet algorithme à (MP) amène à résoudre une suite de problèmes dont la fonction objectif est de la forme donnée par l'équation (4.77) ou bien la minimisation de son opposée. Le problème maître à résoudre à chaque itération est donc un programme linéaire. Dans la figure 4.11, cet algorithme est intégré à la méthode de SEPGCOL standard.

$$\max N - q_n D = \min q_n D - N \quad (4.77)$$

N^* et D^* représentent les valeurs de N et D pour une solution optimale de (MP). À chaque nœud, q_0 peut prendre la valeur 0 ou la valeur d'une solution courante. Le fait de résoudre à chaque itération le problème maître sur un ensemble Ω' restreint, nuit à la convergence de la suite (q_n) . En effet, il n'y a pas forcément une croissance stricte de (q_n) sur Ω' restreint. Cependant, lorsqu'il y a stabilisation de la suite (q_n) , il est *toujours* possible de générer des colonnes de coût réduit négatif dans Ω , car la solution optimale sur Ω est au moins aussi bonne que sur Ω' et que cet optimum (à ϵ près) est atteint il l'aurait été à l'itération précédente puisque la valeur de q_n n'a pas changé. Le fait de générer des colonnes peut, quant à lui, faire diminuer la valeur de q_n ce qui nuit aussi à la convergence. Puisque Ω est convexe et de taille finie, et que dans tous les cas de non-optimalité des colonnes sont générées, on peut conclure que notre algorithme est convergent (à ϵ près).

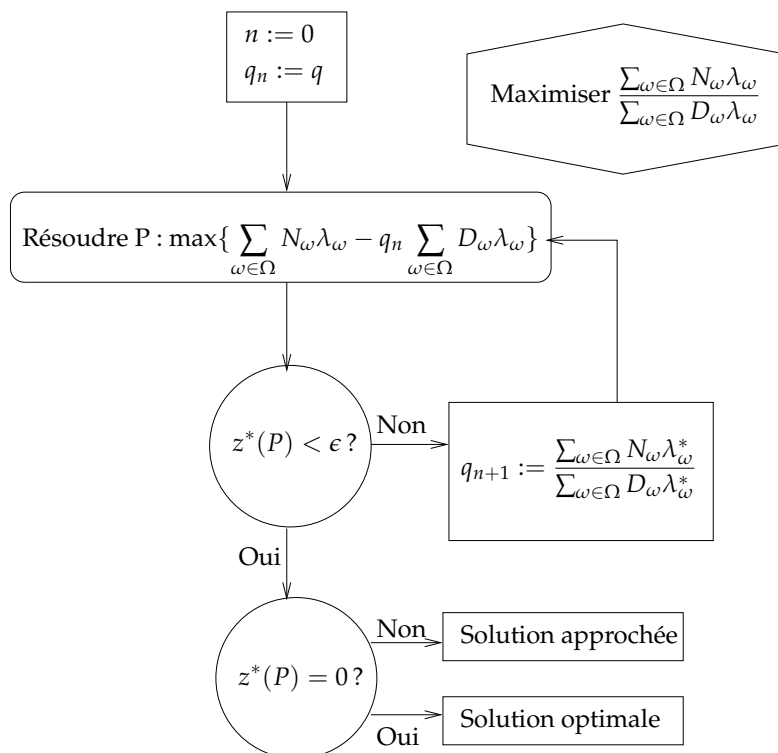


FIG. 4.10: L'algorithme de Dinkelbach

4.2.3.1.2 Résolution du problème esclave À chaque itération de l'algorithme de génération de colonnes, la fonction objectif du problème esclave s'écrit par la formule (4.78) avec π_ρ^i nul pour i une livraison.

$$\min qD_\omega - N_\omega - \sum_{r=1}^R \rho_\omega^r \pi_\rho^r = \min \sum_{(v_i, v_j) \in \bar{\omega}} [(q - f_i)T_{ij} - \pi_\rho^i] \quad (4.78)$$

La fonction d'extension ref_C d'un chemin partiel ch , qui est évidemment croissante, est définie par la relation (4.79).

$$ref_C(ch, (v_i, v_j)) = \begin{cases} C_{ch} + (q - F_{ch})T_{ij} - \pi_\rho^j & \text{si } j \in \mathcal{R}^+ \\ C_{ch} + (q - F_{ch})T_{ij} & \text{sinon} \end{cases} \quad (4.79)$$

Nous cherchons à évaluer $gain_{ij}(k) = C_{ik} + C_{kj} - C_{ij}$, soit le gain associé à un détour par k entre i et j . Ce gain est fonction du gain Δ_{ijk} , défini par l'équation (4.80) et du flux de passagers f_i , après avoir remarqué que $f_k = f_i + F_k$.

$$\Delta_{ijk} = T_{ik} + T_{kj} - T_{ij} \quad (4.80)$$

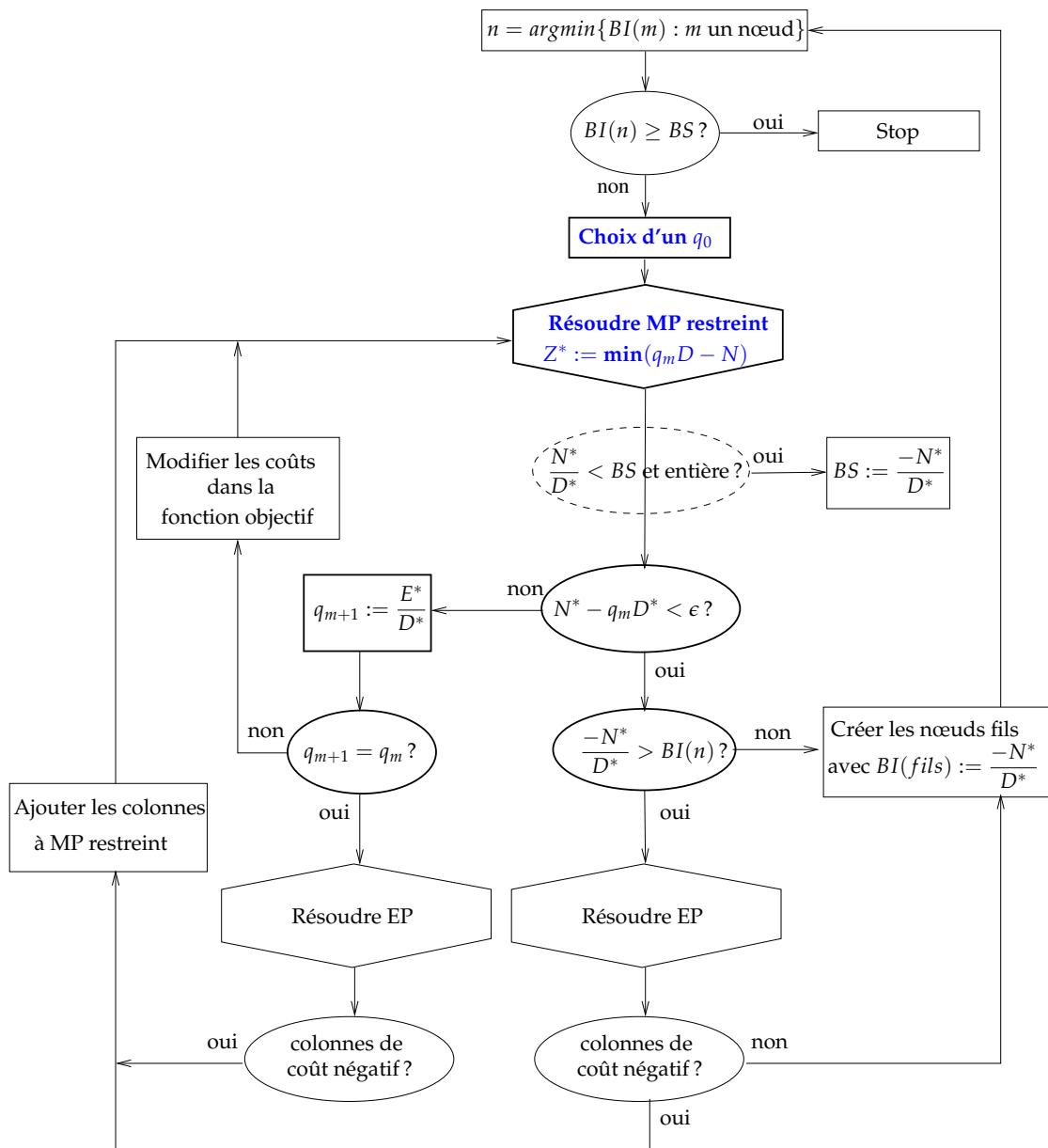


FIG. 4.11: L'algorithme de Dinkelbach avec la génération de colonnes – pour minimiser $-N/D$

La valeur de q représente un taux de remplissage moyen, l'expression $q - f_i$ peut donc prendre des valeurs négatives. D'après cette remarque et la figure 4.12, il est évident que quelque soit la valeur duale associée aux ramassages ou aux livraisons, l'inégalité triangulaire ne sera respectée dans aucun des deux cas (k un ramassage ou une livraison). Par conséquent, la règle de dominance sur les coûts devient invalide car des détours obligatoires peuvent être source de gains pour la fonction objectif. Ceci interdit l'utilisation de la règle de dominance R2 de l'algorithme de programmation dynamique, comme cela est discuté dans la section 4.1.3.2.

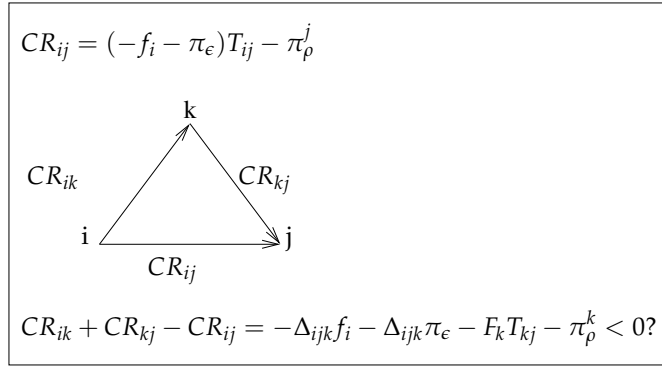


FIG. 4.12: Inégalité triangulaire pour la méthode itérative

L'utilisation de la règle R3, permettant de conserver la règle d'inclusion entre les ensembles de requêtes en cours, nécessite d'estimer le gain potentiel minimal obtenu par des extensions obligatoires ou génératrices du diminution du coût.

Des minoration sur les coûts des détours liés à chaque livraison sont définitivement calculables en prétraitement pour q fixé par la formule (4.81). Calculer $bic_{ij}(k)$ revient à maximiser ou minimiser f_i suivant le signe de Δ_{ijk} , toujours positif lorsque l'inégalité triangulaire est respectée par les durées de transport. Les valeurs maximales des f_i sont données dans les tableaux 4.6 et 4.7 en fonction de la nature des sommets i et j et de la capacité F du véhicule⁴. Nous introduisons la notation $|x|^\oplus = \max\{x, 0\}$.

$$bic(k) = \min_{\{v_i, v_j\} \in \mathcal{V}_\omega^2} (-\Delta_{ijk}f_i + q\Delta_{ijk} - F_k T_{kj} - \pi_\rho^k) \quad (4.81)$$

$k \in \mathcal{R}^-$	$j \in \mathcal{R}^+$	$j \in \mathcal{R}^-$	$j = -R - 1$
$i \in \mathcal{R}^+$	$F - F_j + F_k ^\oplus$	F	$-F_k$
$i \in \mathcal{R}^-$	$F - F_i + F_j + F_k ^\oplus ^\oplus$	$F + F_i$	$-F_k$

TAB. 4.6: Valeurs maximales de f_i pour $k \in \mathcal{R}^-$

$k \in \mathcal{R}^+$	$j \in \mathcal{R}^+$	$j \in \mathcal{R}^-$
$i \in \mathcal{R}^+$	$F - F_k - F_j$	$F - F_k$
$i \in \mathcal{R}^-$	$F - F_i + F_k + F_j ^\oplus$	$F - F_i + F_k ^\oplus$
$i = R + 1$	0	0

TAB. 4.7: Valeurs maximales de f_i pour $k \in \mathcal{R}^+$

En appliquant les règles de dominance de (4.82)-(4.86), l'algorithme 1 de programmation dynamique s'adapte sans plus de difficultés.

⁴Nous rappelons que $F_{r^+} = -F_{r^-}$ et que $\pi_\rho^{k^-} = 0$.

$$et_{ch} = et_{ch'} \quad (4.82)$$

$$T_{ch} \leq T_{ch'} \quad (4.83)$$

$$F_{ch} \leq F_{ch'} \quad (4.84)$$

$$Cour(ch) \subseteq Cour(ch') \quad (4.85)$$

$$C_{ch} \leq C_{ch'} + BIC_1(\mathcal{O}(ch') \setminus \mathcal{O}(ch)) + BIC_2(\mathcal{F}(ch) \setminus \mathcal{UFO}(ch')) \quad (4.86)$$

4.2.3.2 La méthode directe

La méthode directe transforme le problème maître à l'aide d'une opération ressemblant à un changement de variable classique tel que $1 / \sum_{\omega \in \Omega} D_{\omega} \lambda_{\omega} = t$. La fonction objectif maximisant le rapport N/D est remplacée la minimisation de son opposé.

(MPR')

$$\min -t \sum_{\omega \in \Omega} N_{\omega} \lambda_{\omega} \quad (4.87)$$

s.c.q.

$$\left(\sum_{\omega \in \Omega} \rho_{\omega}^r \lambda_{\omega} \right) * t = 1 * t \quad \forall r = 1, \dots, R, \quad [\pi_{\rho}^r] \quad (4.88)$$

$$\sum_{\omega \in \Omega} D_{\omega} \lambda_{\omega} t = 1, \quad [\pi_{\epsilon}] \quad (4.89)$$

$$\lambda_{\omega} t \geq 0 \quad \forall \omega \in \Omega, \quad (4.90)$$

$$t \geq 0. \quad (4.91)$$

En remplaçant $\lambda_{\omega} t$ par λ'_{ω} , on obtient (MPR) qui est un programme linéaire. D'où le schéma de résolution de la figure 4.13.

(MPR)

$$\min \sum_{\omega \in \Omega} -N_{\omega} \lambda'_{\omega} \quad (4.92)$$

s.c.q.

$$\sum_{\omega \in \Omega} \rho_{\omega}^r \lambda'_{\omega} - t = 0 \quad \forall r = 1, \dots, R, \quad [\pi_{\rho}^r] \quad (4.93)$$

$$\sum_{\omega \in \Omega} D_{\omega} \lambda'_{\omega} = 1, \quad [\pi_{\epsilon}] \quad (4.94)$$

$$\lambda'_{\omega} \geq 0 \quad \forall \omega \in \Omega, \quad (4.95)$$

$$t > 0. \quad (4.96)$$

La contrainte supplémentaire (4.94) conséquente au changement de variable, génère une variable duale π_{ϵ} qui doit être intégrée au problème esclave.

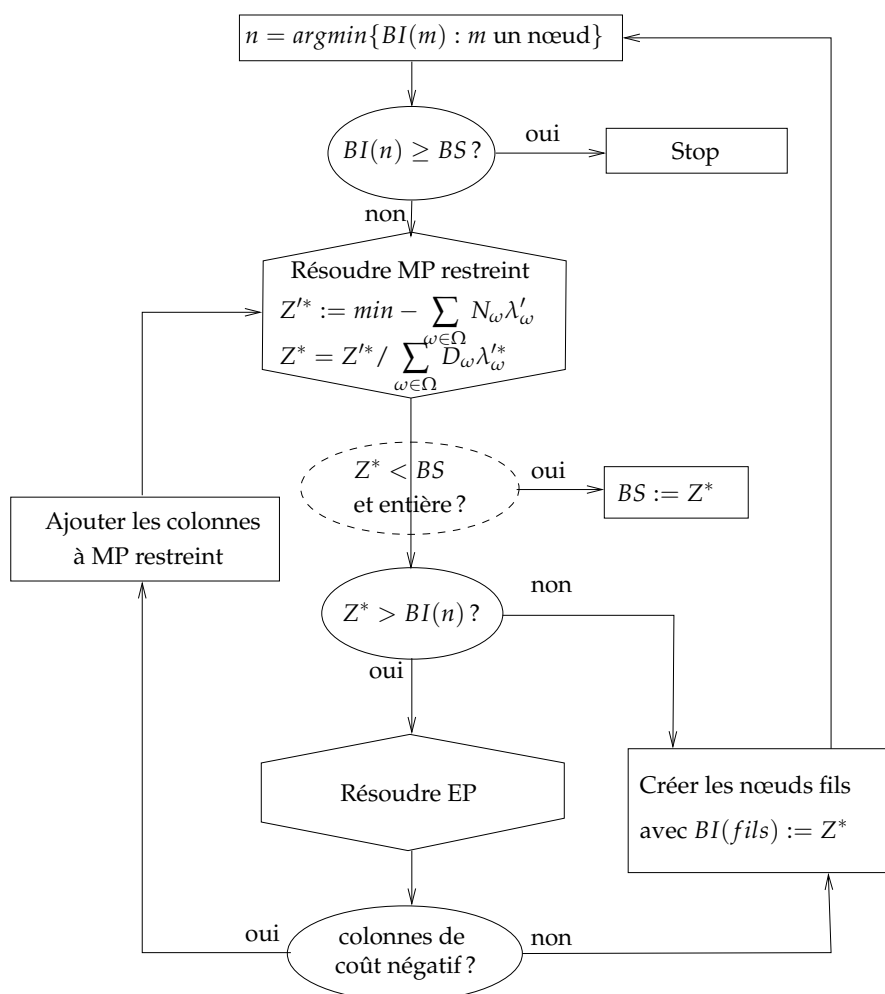


FIG. 4.13: Méthode directe intégrée au SEPGCOL – pour minimiser $-N/D$

4.2.3.2.1 Résolution du problème esclave La fonction objectif du problème esclave prend la forme de l'équation (4.97).

(EP)

$$\min \sum_{(v_i, v_j) \in \mathcal{A}} -[(f_i + \pi_\epsilon)T_{ij} + \pi_\rho^j] \delta_{ij} \quad (4.97)$$

On peut ainsi associer un coût CR_{ij} à chaque arc. On constate, voir figure 4.14, que ces coûts ne respectent pas l'inégalité triangulaire même pour des détours dont la duale associée (π_ρ^k) est nulle. Évaluer à chaque itération (du problème esclave) la valeur minimale du gain pour le triplet (i, j, k) revient à maximiser la valeur du flux de passagers à la sortie de i , soit f_i . Ces valeurs sont les mêmes que celles données dans les tableaux 4.6 et 4.7 pour la méthode itérative.

On en déduit les mêmes règles de dominance que dans le cas itératif avec un calcul des bornes inférieures sur les diminutions du coût des extensions différent. L'équation

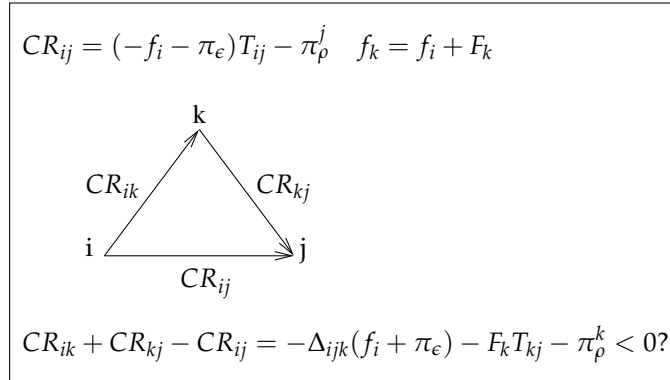


FIG. 4.14: Inégalité triangulaire pour la méthode directe

(4.98) donne l'expression du coût minimal pour les détours par le service k .

$$bic(k) = \min_{(v_i, v_j) \in \mathcal{V}_\omega^2} (-\Delta_{ijk}f_i - \Delta_{ijk}\pi_\epsilon - F_k T_{kj}) - \pi_{rho}^k \quad (4.98)$$

4.2.4 Résultats

Nous présentons dans cette section les résultats obtenus sur les séries d'instances modifiées de la littérature (' gr' ', ' gc' ' et ' grc' ', ' gb' ') présentées dans la section 4.1.4. Nous avons plongé ces instances dans un p -graphe pour générer des séries supplémentaires notées ' mr' ', ' mc' ', ' mrc' ' et ' mb' '. Les p -graphe sont construits en ajoutant itérativement un arc entre deux sommets (service ou dépôt) choisis aléatoirement sans tenir compte des contraintes de fenêtre de temps ou de capacité. Pour les instances présentées ici, $\mathcal{V}^2/2$ arcs sont ajoutés⁵. Les arcs supplémentaires sont pondérés par des valeurs indépendantes comprises entre 120% et 140% pour la durée et 60% et 80% pour la distance, du poids de l'arc d'origine pour lequel la durée égale la distance.

Les chemins non-dominés par rapport à la distance et à la durée sont calculés entre chaque paire de sommets. Les caractéristiques des instances ainsi obtenues sont présentées dans les tableaux 4.8 et 4.9. On constate que la densité du p -graphe (en gras) est entre 4 et 10 fois supérieure à celle du 1-graphe pour les instances ' mr' ' et ' mrc' ' et entre 15 et 18 fois supérieure pour les instances ' mc' '.

Les conditions expérimentales sont identiques à celles des résultats obtenus sur les instances ' g' ' de la section 4.1.4, hormis pour le temps perdu où la taille de la flotte est bornée par la valeur obtenue dans la solution optimale calculée par rapport à la minimisation de la distance totale parcourue (dans un 1-graphe). Les tests sont effectués suivant les trois critères énoncés sur les instances ' g' '. Les résultats se trouvent dans les tableaux 4.10 et 4.11 pour les instances ' gr' ' (uniformément distribuées), le tableau 4.12

⁵Nous rappelons que le graphe d'origine est un graphe complet et que les arcs correspondent aux distances euclidiennes.

nom	req.	largeur	écart	densité p -G	dens, 1-G	flux
mr101	53	5,72	19,06	0,3708	0,0485	1,91
mr102	55	5,70	19,23	0,6471	0,0687	2,13
mr103	52	6,41	21,16	0,6429	0,0735	1,92
mr104	52	5,77	19,19	0,6498	0,0659	1,88
mr105	53	5,49	18,61	0,4373	0,0524	2,13
mr106	52	6,17	20,57	0,4473	0,0602	2,08
mr107	52	7,21	23,51	0,6816	0,0715	1,88
mr108	50	6,99	22,89	0,5985	0,0642	1,92
mr109	53	6,77	22,34	0,3318	0,0460	2,02
mr110	52	5,82	19,41	0,3124	0,0453	1,96
mr111	54	5,66	18,74	0,4342	0,0532	1,76
mr112	53	6,41	21,20	0,0846	0,0239	1,96
mc101	53	2,66	10,86	1,5778	0,1120	2,87
mc102	53	2,84	11,54	1,7725	0,1139	3,02
mc103	52	3,00	11,82	1,7372	0,1133	2,81
mc104	53	3,18	12,21	1,3188	0,0873	2,66
mc105	53	2,63	10,83	1,6691	0,1076	2,94
mc106	53	2,76	11,10	1,6752	0,1054	2,83
mc107	53	2,86	11,48	1,8544	0,1038	2,91
mc108	53	2,84	11,61	1,5326	0,1005	3,08
mc109	53	2,62	10,68	1,6463	0,0976	2,83
mrc101	53	6,18	20,95	0,1834	0,0351	2,40
mrc102	53	6,90	23,20	0,4815	0,0571	2,51
mrc103	53	5,51	18,97	0,6284	0,0672	2,43
mrc104	54	5,84	19,81	0,4937	0,0551	2,28
mrc105	54	6,74	22,68	0,2273	0,0380	2,44
mrc106	53	6,80	22,72	0,2244	0,0395	2,32
mrc107	53	6,05	20,65	0,1691	0,0323	2,49
mrc108	52	6,45	21,76	0,0818	0,0245	2,40

TAB. 4.8: Caractéristiques des instances de Li et Lim modifiées par un p -graphe

pour les instances 'gc' (groupées), le tableau 4.13 pour les instances 'grc' (mixtes) et les tableaux 4.14, 4.15 et 4.16 pour les instances 'gb'. Pour chaque instance de référence, nous donnons les résultats de cinq optimisations : suivant la distance totale parcourue dans un 1-graphe puis dans un p -graphe, le temps perdu pour des requêtes de type *aller* et le taux de remplissage (en fait son inverse) par la méthode directe puis itérative. La valeur obtenue pour le critère optimisé est indiquée en gras. Afin de mieux cerner le type de solutions produites, nous avons rajouté deux indications, la durée totale des tournées (dans la colonne *durée*) et les durées pendant lesquelles les véhicule sont en déplacement (dans la colonne *temps roulé*). À la fin de chaque tableau nous affichons les résultats moyens sur les instances du tableau (exception faite du tableau 4.10 qui est intégré dans la moyenne du tableau 4.11).

La première observation valable pour toutes les instances, est la qualité de la relaxation linéaire du problème maître ; la valeur optimale de cette dernière étant presque toujours égale à celle du problème entier ($MP\% \approx 0,0$) et une solution entière étant fournie au nœud racine de l'arbre de recherche. L'apport des chemins alternatifs est très impor-

4.2. Intégration des critères de qualité de service

nom	veh.	largeur	écart	dens. p-g.	dens. l-g.	flux
gb2-16	16	5,50	16,50	0,7412	0,1289	3,00
gb2-20	20	4,70	14,20	0,9594	0,1263	4,20
gb2-24	24	5,70	17,00	0,8689	0,1272	3,60
gb3-18	18	5,30	16,00	0,7338	0,1227	3,80
gb3-24	24	4,70	14,20	0,9080	0,1207	3,60
gb3-30	30	5,70	17,10	1,0639	0,1233	3,60
gb3-36	36	5,30	16,00	1,2218	0,1233	3,70
gb4-16	16	6,00	18,10	0,6865	0,1221	4,40
gb4-24	24	4,50	13,60	0,6766	0,1198	3,60
gb4-32	32	5,10	15,20	1,0930	0,1240	3,80
gb4-40	40	5,20	15,60	1,1678	0,1222	4,00
gb4-48	48	4,80	14,30	1,2195	0,1233	3,30
gb5-40	40	5,70	17,00	1,3150	0,1213	3,20
gb5-50	50	5,50	16,60	1,4496	0,1209	3,60
gb5-60	60	5,40	16,30	1,7731	0,1203	3,20
gb6-48	48	5,50	16,50	1,4902	0,1198	3,80
gb6-60	60	5,50	16,50	1,9451	0,1217	3,60
gb6-72	72	5,20	15,60	1,8971	0,1223	3,50
gb7-56	56	5,50	16,40	1,6917	0,1196	3,60
gb7-70	70	4,60	13,90	1,8830	0,1207	3,80
gb7-84	84	5,70	17,10	2,1428	0,1215	3,40
gb8-64	64	4,80	14,50	1,8862	0,1201	3,60
gb8-80	80	5,00	15,00	1,9986	0,1208	3,80
gb8-96	96	5,00	15,00	2,3407	0,1217	3,30

TAB. 4.9: Caractéristiques des instances de Cordeau modifiées par un p-graphe

nom	veh.	dist.	tps. perdu	tx remp.	durée	tps roulé	sec.	MP%	nœu.
gr101	24,0	1988,5	38,9	1,7568	2645,3	1988,5	0,0	0,0	1,0
	25,0	1530,3	167,7	1,9082	3243,0	2569,0	9,0	0,1	3,0
	24,0	2238,8	27,5	1,9853	2803,0	2238,8	0,0	0,0	1,0
	25,0	2001,7	49,2	1,7479	2644,9	2001,7	0,0	0,1	3,0
	25,0	2001,7	49,2	1,7479	2644,9	2001,7	0,0	0,1	3,0
gr102	22,0	1927,2	30,1	1,4128	2987,1	1927,2	1,0	0,0	1,0
	23,0	1451,6	167,4	1,5181	3499,6	2528,0	20,0	0,0	1,0
	22,0	2210,5	4,6	1,6477	3049,1	2210,5	1,0	0,0	5,0
	22,0	1927,2	30,1	1,4128	2987,1	1927,2	0,0	0,0	1,0
	22,0	1927,2	30,1	1,4128	2987,1	1927,2	6,0	0,0	1,0
gr103	17,0	1812,9	49,9	1,3086	2871,1	1812,9	1,0	0,0	1,0
	17,0	1398,4	196,4	1,4021	3264,7	2321,3	17,0	0,0	1,0
	17,0	2286,9	0,1	1,7056	3063,9	2286,9	0,0	0,0	7,0
	17,0	1812,9	49,9	1,3086	2871,1	1812,9	0,0	0,0	1,0
	17,0	1812,9	49,9	1,3086	2871,1	1812,9	6,0	0,0	1,0
gr104	25,0	1897,6	60,1	1,5725	3108,5	1897,6	0,0	0,0	1,0
	25,0	1440,6	145,3	1,7967	3445,7	2429,0	8,0	0,0	1,0
	25,0	2432,6	27,2	2,0769	3178,5	2432,6	0,0	0,0	2,0
	25,0	1897,6	60,1	1,5725	3108,5	1897,6	0,0	0,0	1,0
	25,0	1897,6	60,1	1,5725	3108,5	1897,6	1,0	0,0	1,0
gr105	25,0	2046,4	30,0	1,6570	2717,5	2046,4	0,0	0,0	1,0
	25,0	1562,8	148,5	1,8228	3221,7	2658,3	8,0	0,0	1,0
	25,0	2352,0	1,0	1,9085	2778,8	2352,0	0,0	0,0	1,0
	25,0	2046,4	30,0	1,6570	2717,5	2046,4	0,0	0,0	1,0
	25,0	2046,4	30,0	1,6570	2717,5	2046,4	1,0	0,0	1,0
gr106	18,0	1864,0	54,7	1,3528	2445,0	1864,0	0,0	0,0	1,0
	20,0	1426,3	179,1	1,4272	2980,8	2398,5	12,0	0,0	1,0
	18,0	2087,4	38,0	1,5149	2588,7	2087,4	0,0	0,0	6,0
	18,0	1864,0	54,7	1,3528	2445,0	1864,0	0,0	0,0	1,0
	18,0	1864,0	54,7	1,3528	2445,0	1864,0	4,0	0,0	1,0

TAB. 4.10: Résultats pour les instances de Li et Lim modifiées – gr101-gr106

4.2. Intégration des critères de qualité de service

nom	veh.	dist.	tps. perdu	tx remp.	durée	tps roulé	sec.	MP%	nœu.
gr107	17,0	1834,5	46,6	1,2941	2902,9	1834,5	0,0	0,0	1,0
	19,0	1412,5	201,7	1,3863	3516,7	2390,7	14,0	0,0	1,0
	17,0	2201,5	18,0	1,5531	3128,6	2201,5	0,0	0,0	3,0
	17,0	1834,5	46,6	1,2941	2902,9	1834,5	0,0	0,0	1,0
	17,0	1834,5	46,6	1,2941	2902,9	1834,5	3,0	0,0	1,0
gr108	27,0	2125,7	28,0	1,5958	3339,4	2125,7	0,0	0,0	1,0
	27,0	1588,5	190,1	1,6763	3782,7	2729,3	4,0	0,0	1,0
	27,0	2484,1	27,4	1,8649	3287,3	2484,1	0,0	0,0	1,0
	27,0	2130,0	36,5	1,5937	3339,5	2130,0	0,0	0,0	1,0
	27,0	2130,0	36,5	1,5937	3339,5	2130,0	1,0	0,0	1,0
gr109	23,0	1917,8	65,2	1,2567	2509,5	1917,8	0,0	0,0	1,0
	23,0	1451,5	227,8	1,3243	2882,5	2405,8	7,0	0,0	1,0
	23,0	2128,5	17,2	1,4331	2576,7	2128,5	0,0	0,0	1,0
	23,0	1917,8	65,2	1,2567	2509,5	1917,8	0,0	0,0	1,0
	23,0	1917,8	65,2	1,2567	2509,5	1917,8	0,0	0,0	1,0
gr110	25,0	1961,4	51,4	1,6632	2587,9	1961,4	0,0	0,0	1,0
	25,0	1476,7	154,4	1,8725	3043,8	2486,9	5,0	0,0	1,0
	25,0	2191,0	30,5	1,8827	2675,9	2191,0	0,0	0,0	1,0
	25,0	1963,7	58,4	1,6473	2589,3	1963,7	0,0	0,0	1,0
	25,0	1963,7	58,4	1,6473	2589,3	1963,7	0,0	0,0	1,0
gr111	21,0	1844,0	56,9	1,6298	2460,4	1844,0	0,0	0,6	3,0
	22,0	1413,0	156,9	1,7620	2826,9	2347,3	16,0	0,8	3,0
	21,0	2053,6	33,4	1,8245	2599,8	2053,6	0,0	0,0	5,0
	21,0	1844,0	56,9	1,6298	2460,4	1844,0	0,0	0,3	3,0
	21,0	1844,0	56,9	1,6298	2460,4	1844,0	2,0	0,3	3,0
gr112	27,0	2182,2	78,9	1,6704	2444,5	2182,2	0,0	0,0	1,0
	27,0	1705,3	171,4	1,8694	2976,9	2719,5	1,0	0,0	1,0
	27,0	2224,1	67,1	1,7156	2496,7	2224,1	0,0	0,0	1,0
	28,0	2196,4	92,5	1,6600	2438,8	2196,4	0,0	0,0	1,0
	28,0	2196,4	92,5	1,6600	2438,8	2196,4	0,0	0,0	1,0
Moyenne	22,6	1950,2	49,2	1,5142	2751,6	1950,2	0,2	0,1	1,2
	23,2	1488,1	175,6	1,6472	3223,8	2498,6	10,1	0,1	1,3
	22,6	2240,9	24,3	1,7594	2852,3	2240,9	0,1	0,0	2,8
	22,8	1953,0	52,5	1,5111	2751,2	1953,0	0,0	0,0	1,3
	22,8	1953,0	52,5	1,5111	2751,2	1953,0	2,0	0,0	1,3

TAB. 4.11: Résultats pour les instances de Li et Lim modifiées – gr107-gr112

Chapitre 4. Maximisation de la qualité de service

nom	veh.	dist.	tps. perdu	tx remp.	durée	tps roulé	sec.	MP%	nœu.
gc101	9,0	1002,8	4,1	1,2251	6085,2	1002,8	5,0	0,0	1,0
	9,0	795,6	62,8	1,2918	6392,4	1284,7	225,0	0,0	1,0
	9,0	2158,5	0,0	2,6370	6240,0	2158,5	5,0	0,0	18,0
	9,0	1002,8	4,1	1,2251	6085,2	1002,8	4,0	0,0	1,0
	9,0	1002,8	4,1	1,2251	6085,2	1002,8	2,0	0,0	1,0
gc102	9,0	1060,9	0,1	1,1274	8063,4	1060,9	4,0	0,0	1,0
	9,0	813,4	78,7	1,1682	8244,1	1383,6	413,0	0,0	1,0
	9,0	2037,6	0,0	2,1653	6791,4	2037,6	4,0	0,0	13,0
	9,0	1060,9	0,1	1,1274	8063,4	1060,9	6,0	0,0	1,0
	9,0	1060,9	0,1	1,1274	8063,4	1060,9	3,0	0,0	1,0
gc103	13,0	1250,4	0,0	1,4531	11136,1	1250,4	3,0	0,0	1,0
	13,0	942,8	75,6	1,5538	12999,2	1645,7	168,0	0,0	1,0
	13,0	2056,6	0,0	2,3899	10747,5	2056,6	3,0	0,0	10,0
	13,0	1250,4	0,0	1,4531	11136,1	1250,4	3,0	0,0	1,0
	13,0	1250,4	0,0	1,4531	11136,1	1250,4	2,0	0,0	1,0
gc104	28,0	1982,2	10,8	2,2178	12631,2	1982,2	0,0	0,0	1,0
	28,0	1438,4	71,1	2,5096	13234,6	2633,1	43,0	0,0	1,0
	28,0	2569,7	10,3	2,8751	7070,9	2569,7	0,0	0,0	9,0
	28,0	1982,2	10,8	2,2178	12631,2	1982,2	1,0	0,0	1,0
	28,0	1982,2	10,8	2,2178	12631,2	1982,2	1,0	0,0	1,0
gc105	11,0	1185,4	2,8	1,3680	6935,7	1185,4	3,0	0,0	1,0
	11,0	900,1	60,1	1,5155	7363,4	1584,7	417,0	0,0	1,0
	11,0	1801,4	0,0	2,0789	5627,3	1801,4	1,0	0,0	4,0
	11,0	1185,4	2,8	1,3680	6935,7	1185,4	4,0	0,0	1,0
	11,0	1185,4	2,8	1,3680	6935,7	1185,4	2,0	0,0	1,0
gc106	13,0	1244,7	4,1	1,5093	6851,7	1244,7	2,0	0,0	1,0
	13,0	952,5	73,4	1,6320	7237,4	1636,7	328,0	0,0	1,0
	13,0	1748,5	0,0	2,1202	6719,0	1748,5	2,0	0,0	1,0
	13,0	1244,7	4,1	1,5093	6851,7	1244,7	4,0	0,0	1,0
	13,0	1244,7	4,1	1,5093	6851,7	1244,7	3,0	0,0	1,0
gc107	13,0	1305,9	15,0	1,4353	7478,0	1305,9	3,0	0,0	1,0
	14,0	982,5	70,6	1,6377	7999,6	1725,8	292,0	0,0	1,0
	13,0	1658,8	14,0	1,8232	6966,0	1658,8	1,0	0,0	1,0
	13,0	1305,9	15,0	1,4353	7478,0	1305,9	4,0	0,0	1,0
	13,0	1305,9	15,0	1,4353	7478,0	1305,9	1,0	0,0	1,0
gc108	12,0	1191,3	1,4	1,2521	6030,7	1191,3	3,0	0,0	1,0
	12,0	941,8	75,3	1,3257	6588,9	1562,8	228,0	0,0	1,0
	12,0	1671,4	0,0	1,7568	5946,4	1671,4	3,0	0,0	9,0
	12,0	1191,3	1,4	1,2521	6030,7	1191,3	3,0	0,0	1,0
	12,0	1191,3	1,4	1,2521	6030,7	1191,3	2,0	0,0	1,0
gc109	17,0	1453,2	0,7	1,7678	6927,4	1453,2	2,0	0,0	1,0
	17,0	1116,6	56,7	1,8993	6807,0	1842,2	231,0	0,0	1,0
	17,0	2255,8	0,4	2,7444	6957,9	2255,8	1,0	0,0	5,0
	17,0	1453,2	0,7	1,7678	6927,4	1453,2	3,0	0,0	1,0
	17,0	1453,2	0,7	1,7678	6927,4	1453,2	1,0	0,0	1,0
Moyenne	13,9	1297,4	4,3	1,4840	8015,5	1297,4	2,8	0,0	1,0
	14,0	987,1	69,4	1,6148	8540,7	1699,9	260,6	0,0	1,0
	13,9	1995,4	2,7	2,2879	7007,4	1995,4	2,2	0,0	7,8
	13,9	1297,4	4,3	1,4840	8015,5	1297,4	3,6	0,0	1,0
	13,9	1297,4	4,3	1,4840	8015,5	1297,4	1,9	0,0	1,0

TAB. 4.12: Résultats pour les instances de Li et Lim modifiées – gc101-gc109

4.2. Intégration des critères de qualité de service

nom	veh.	dist.	tps. perdu	tx remp.	durée	tps roulé	sec.	MP%	nœu.
grc101	22,0	2290,9	39,1	1,4907	2756,7	2290,9	0,0	0,0	1,0
	22,0	1769,1	144,2	1,6395	3388,3	2917,3	5,0	0,4	15,0
	22,0	2433,7	36,2	1,5836	2874,8	2433,7	0,0	0,0	1,0
	22,0	2297,7	64,4	1,4351	2761,5	2297,7	0,0	0,5	5,0
	22,0	2297,7	64,4	1,4351	2761,5	2297,7	0,0	0,5	5,0
grc102	19,0	2145,7	112,2	1,1575	3034,7	2145,7	0,0	0,0	1,0
	21,0	1645,8	232,8	1,2544	3610,9	2749,4	12,0	0,0	1,0
	19,0	2500,2	78,8	1,3520	3189,4	2500,2	0,0	0,0	1,0
	20,0	2149,7	114,8	1,1479	3017,0	2149,7	0,0	0,0	1,0
	20,0	2149,7	114,8	1,1479	3017,0	2149,7	2,0	0,0	1,0
grc103	17,0	2055,4	36,0	1,4318	3215,1	2055,4	0,0	0,0	1,0
	18,0	1545,0	166,7	1,5242	3752,1	2666,1	12,0	0,0	1,0
	17,0	2347,4	29,0	1,6352	3294,9	2347,4	0,0	0,0	4,0
	17,0	2055,4	36,0	1,4318	3215,1	2055,4	0,0	0,0	1,0
	17,0	2055,4	36,0	1,4318	3215,1	2055,4	2,0	0,0	1,0
grc104	27,0	2599,8	45,0	1,8858	3571,9	2599,8	0,0	0,0	1,0
	27,0	1898,6	177,1	2,0443	4177,5	3324,1	4,0	0,0	1,0
	27,0	2889,0	39,2	2,0955	3548,1	2889,0	0,0	0,0	3,0
	27,0	2602,1	50,6	1,8669	3532,7	2602,1	0,0	0,0	1,0
	27,0	2602,1	50,6	1,8669	3532,7	2602,1	1,0	0,0	1,0
grc105	25,0	2543,2	60,6	1,6172	3097,7	2543,2	0,0	0,0	1,0
	25,0	1856,4	239,0	1,6930	3803,3	3313,3	3,0	0,0	1,0
	25,0	2762,5	24,2	1,7672	3270,5	2762,5	0,0	0,0	1,0
	25,0	2544,8	66,6	1,6111	3099,4	2544,8	0,0	0,0	1,0
	25,0	2544,8	66,6	1,6111	3099,4	2544,8	0,0	0,0	1,0
grc106	20,0	2179,9	100,7	1,2627	2736,5	2179,9	0,0	0,0	1,0
	21,0	1693,0	187,7	1,4191	3230,1	2714,3	5,0	0,2	7,0
	20,0	2270,7	74,5	1,3244	2701,2	2270,7	0,0	0,0	1,0
	20,0	2189,7	97,0	1,2550	2730,7	2189,7	0,0	0,0	1,0
	20,0	2189,7	97,0	1,2550	2730,7	2189,7	0,0	0,0	1,0
grc107	29,0	2632,9	75,0	1,6978	3170,4	2632,9	0,0	0,0	1,0
	29,0	1926,9	180,0	1,9019	3844,7	3377,1	2,0	0,0	1,0
	29,0	2864,7	31,0	1,8707	3264,9	2864,7	0,0	0,0	2,0
	29,0	2635,2	89,4	1,6830	3172,6	2635,2	0,0	0,0	1,0
	29,0	2635,2	89,4	1,6830	3172,6	2635,2	0,0	0,0	1,0
grc108	30,0	2662,9	57,8	1,6981	3124,5	2662,9	0,0	0,0	1,0
	30,0	1965,5	159,9	1,8695	3774,4	3397,1	1,0	0,0	1,0
	30,0	2800,8	25,9	1,8175	3202,7	2800,8	0,0	0,0	1,0
	30,0	2675,9	85,0	1,6579	3139,7	2675,9	0,0	0,0	1,0
	30,0	2675,9	85,0	1,6579	3139,7	2675,9	0,0	0,0	1,0
Moyenne	23,6	2388,8	65,8	1,5302	3088,4	2388,8	0,0	0,0	1,0
	24,1	1787,5	185,9	1,6682	3697,7	3057,3	5,5	0,1	3,5
	23,6	2608,6	42,4	1,6808	3168,3	2608,6	0,0	0,0	1,8
	23,8	2393,8	75,5	1,5111	3083,6	2393,8	0,0	0,1	1,5
	23,8	2393,8	75,5	1,5111	3083,6	2393,8	0,6	0,1	1,5

TAB. 4.13: Résultats pour les instances de Li et Lim modifiées – grc101-grc108

nom	veh.	dist.	tps perdu	tx remp.	durée	tps roulé	sec.	MP%	nce.
gb2-16	5	294,3	7,6	0,5289	1107,4	294,3	0	0,0	1
	4	221,4	59,9	0,5451	1038,6	381,7	19	0,0	1
	3	338,1	0,0	0,6198	1016,6	338,1	0	0,0	5
	5	294,3	7,6	0,5289	1107,4	294,3	0	0,0	1
	5	294,3	7,6	0,5289	1107,4	294,3	0	0,0	1
gb2-20	6	314,0	0,0	0,4001	2059,4	314,0	0	0,0	1
	6	226,4	59,7	0,3998	2099,6	411,0	19	0,0	1
	6	366,0	0,0	0,4664	1249,0	366,0	0	0,0	2
	5	315,9	6,2	0,3908	1943,7	315,9	1	0,0	1
	5	315,9	6,2	0,3908	1943,7	315,9	0	0,0	1
gb2-24	4	443,7	0,0	0,4715	1676,9	443,7	0	0,0	1
	5	311,4	93,9	0,4724	1857,6	591,3	22	0,0	1
	4	513,5	0,0	0,5456	1776,6	513,5	1	0,0	10
	4	443,7	0,0	0,4715	1676,9	443,7	1	0,0	1
	4	443,7	0,0	0,4715	1676,9	443,7	1	0,0	1
gb3-18	3	309,9	12,4	0,4183	660,4	309,9	0	0,0	1
	4	233,7	68,1	0,4387	786,4	420,6	20	0,0	1
	3	372,5	1,8	0,5059	741,8	372,5	0	0,0	3
	3	309,9	12,4	0,4183	660,4	309,9	0	0,0	1
	3	309,9	12,4	0,4183	660,4	309,9	0	0,0	1
gb3-24	6	369,7	4,4	0,4523	1705,8	369,7	0	0,0	1
	6	270,0	67,7	0,4628	1656,6	484,7	20	0,0	1
	6	450,0	0,0	0,5506	1542,6	450,0	0	0,0	5
	6	369,7	4,4	0,4523	1705,8	369,7	1	0,0	1
	6	369,7	4,4	0,4523	1705,8	369,7	0	0,0	1
gb3-30	7	524,4	3,2	0,3983	2682,7	524,4	1	0,0	1
	7	371,0	122,6	0,3978	2741,4	706,2	25	0,0	1
	7	640,3	0,0	0,4863	2360,5	640,3	0	0,0	7
	7	524,4	3,2	0,3983	2682,7	524,4	2	0,0	1
	7	524,4	3,2	0,3983	2682,7	524,4	0	0,0	1
gb3-36	6	600,5	0,0	0,4140	3352,8	600,5	2	0,0	1
	6	423,0	125,4	0,4080	3388,0	784,9	62	0,0	1
	6	758,1	0,0	0,5227	2637,2	758,1	2	0,0	10
	6	600,5	0,0	0,4140	3352,8	600,5	5	0,0	1
	6	600,5	0,0	0,4140	3352,8	600,5	2	0,0	1
gb4-16	4	314,3	4,2	0,3670	689,8	314,3	0	0,0	1
	4	232,8	62,1	0,3714	726,4	414,1	19	0,0	1
	4	376,5	0,0	0,4395	600,8	376,5	0	0,0	4
	4	314,5	7,1	0,3650	689,8	314,5	0	0,0	1
	4	314,5	7,1	0,3650	689,8	314,5	0	0,0	1
Moyenne	5,1	396,4	4,0	0,4313	1741,9	396,4	0,4	0,0	1,0
	5,3	286,2	82,4	0,4370	1786,8	524,3	25,8	0,0	1,0
	4,9	476,9	0,2	0,5171	1490,6	476,9	0,4	0,0	5,8
	5,0	396,6	5,1	0,4299	1727,4	396,6	1,3	0,0	1,0
	5,0	396,6	5,1	0,4299	1727,4	396,6	0,4	0,0	1,0

TAB. 4.14: Résultats des 3 critères de QoS pour instances de Cordeau modifiée -1

4.2. Intégration des critères de qualité de service

nom	veh.	dist.	tps perdu	tx remp.	durée	tps roulé	sec.	MP%	noe.
gb4-24	6	395,5	2,9	0,5042	1345,0	395,5	0	0,0	1
	7	300,8	60,2	0,5199	1469,4	510,7	19	0,0	1
	6	488,2	0,0	0,6224	935,0	488,2	0	0,0	4
	6	395,5	2,9	0,5042	1345,0	395,5	1	0,0	1
	6	395,5	2,9	0,5042	1345,0	395,5	0	0,0	1
gb4-32	7	515,6	14,9	0,4216	2523,3	515,6	2	0,0	1
	7	363,0	113,2	0,4364	2417,7	703,6	34	0,0	1
	7	684,0	0,0	0,5695	1857,6	684,0	1	0,0	5
	7	515,6	14,9	0,4216	2523,3	515,6	3	0,0	1
	7	515,6	14,9	0,4216	2523,3	515,6	1	0,0	1
gb4-40	8	653,7	12,0	0,4018	3479,8	653,7	3	0,0	1
	7	460,7	137,2	0,4123	3161,4	870,5	74	0,0	1
	8	790,4	0,0	0,4900	3227,9	790,4	1	0,0	13
	8	653,7	12,0	0,4018	3479,8	653,7	8	0,0	1
	8	653,7	12,0	0,4018	3479,8	653,7	2	0,0	1
gb4-48	9	695,6	30,1	0,4581	4552,6	695,6	10	0,0	1
	8	504,2	156,0	0,4763	4587,5	939,0	181	0,0	1
	9	963,2	0,0	0,6501	3025,5	963,2	5	0,0	7
	9	695,6	30,1	0,4581	4552,6	695,6	19	0,0	1
	9	695,6	30,1	0,4581	4552,6	695,6	10	0,0	1
gb5-40	7	676,9	17,6	0,4577	2390,1	676,9	4	0,0	1
	7	478,1	154,1	0,4646	2357,2	885,1	80	0,0	1
	7	902,7	0,0	0,6219	2010,6	902,7	2	0,0	17
	7	676,9	17,6	0,4577	2390,1	676,9	6	0,0	1
	7	676,9	17,6	0,4577	2390,1	676,9	3	0,0	1
gb5-50	8	832,4	27,5	0,4277	3566,6	832,4	14	0,0	1
	10	589,2	173,2	0,4449	4018,5	1116,0	230	0,0	1
	8	1052,6	0,0	0,5437	3397,9	1052,6	5	0,0	17
	8	832,4	27,5	0,4277	3566,6	832,4	16	0,0	1
	8	832,4	27,5	0,4277	3566,6	832,4	4	0,0	1
gb5-60	10	1000,2	9,7	0,4552	5491,5	1000,2	29	0,0	1
	10	687,4	208,2	0,4587	5484,3	1331,3	644	0,0	1
	10	1290,6	0,0	0,5874	3720,7	1290,6	5	0,0	17
	10	1000,2	9,7	0,4552	5491,5	1000,2	42	0,0	1
	10	1000,2	9,7	0,4552	5491,5	1000,2	5	0,0	1
gb6-48	8	773,5	12,9	0,3637	2730,0	773,5	10	0,0	1
	9	534,5	177,9	0,3687	3094,4	1030,3	211	0,0	1
	8	945,1	0,0	0,4464	2707,9	945,1	4	0,0	14
	8	773,5	12,9	0,3637	2730,0	773,5	15	0,0	1
	8	773,5	12,9	0,3637	2730,0	773,5	2	0,0	1
Moyenne	7,9	692,9	16,0	0,4363	3259,9	692,9	9,0	0,0	1,0
	8,1	489,7	147,5	0,4477	3323,8	923,3	184,1	0,0	1,0
	7,9	889,6	0,0	0,5664	2610,4	889,6	2,9	0,0	11,8
	7,9	692,9	16,0	0,4363	3259,9	692,9	13,8	0,0	1,0
	7,9	692,9	16,0	0,4363	3259,9	692,9	3,4	0,0	1,0

TAB. 4.15: Résultats des 3 critères de QoS pour instances de Cordeau modifiée –2

nom	veh.	dist.	tps perdu	tx remp.	durée	tps roulé	sec.	MP%	noe.
gb6-60	9	935,1	22,0	0,3709	3762,5	935,1	30	0,0	1
	10	639,4	224,2	0,3750	4066,7	1249,2	1187	0,0	1
	9	1262,3	0,0	0,5021	3054,0	1262,3	10	0,0	16
	9	937,0	23,3	0,3705	3762,5	937,0	79	0,0	1
	9	937,0	23,3	0,3705	3762,5	937,0	5	0,0	1
gb6-72	10	1058,4	16,6	0,4109	5438,0	1058,4	76	0,0	1
	11	719,9	244,1	0,4190	6151,2	1412,4	1925	0,0	1
	9	1442,1	0,0	0,5617	3543,9	1442,1	20	0,0	33
	10	1058,4	16,6	0,4109	5438,0	1058,4	115	0,0	1
	10	1058,4	16,6	0,4109	5438,0	1058,4	15	0,0	1
gb7-56	11	924,1	25,2	0,4039	3008,1	924,1	14	0,0	1
	11	633,3	196,2	0,4141	3375,4	1237,7	287	0,0	1
	11	1117,0	0,0	0,4899	2510,6	1117,0	3	0,0	11
	11	925,3	32,8	0,4032	3008,1	925,3	26	0,0	1
	11	925,3	32,8	0,4032	3008,1	925,3	30	0,0	1
gb7-70	12	986,9	8,1	0,4238	4685,2	986,9	48	0,0	1
	12	677,1	197,9	0,4326	4789,5	1305,1	1249	0,0	1
	12	1384,2	0,0	0,5945	3565,6	1384,2	19	0,0	34
	12	988,7	21,2	0,4219	4584,2	988,7	75	0,0	1
	12	988,7	21,2	0,4219	4584,2	988,7	15	0,0	1
gb7-84	12	1323,2	52,0	0,4080	6688,2	1323,2	167	0,0	1
	13	903,6	326,9	0,4117	7493,0	1757,1	4172	0,0	1
	12	1702,2	0,0	0,5296	5379,6	1702,2	26	0,0	24
	12	1324,8	68,8	0,4062	6639,5	1324,8	199	0,0	1
	12	1324,8	68,8	0,4062	6639,5	1324,8	35	0,0	1
gb8-64	11	948,3	15,7	0,4218	3571,9	948,3	38	0,0	1
	11	645,8	206,1	0,4329	3773,0	1275,9	1475	0,0	1
	11	1283,7	0,0	0,5777	3401,1	1283,7	7	0,0	11
	12	951,8	18,7	0,4197	3930,1	951,8	53	0,0	1
	12	951,8	18,7	0,4197	3930,1	951,8	50	0,0	1
gb8-80	10	1132,8	28,7	0,3720	4478,1	1132,8	134	0,0	1
	12	782,4	257,7	0,3805	5108,7	1531,0	3421	0,0	1
	10	1615,1	0,0	0,5305	3962,1	1615,1	57	0,0	31
	10	1133,6	32,1	0,3705	4478,1	1133,6	181	0,0	1
	10	1133,6	32,1	0,3705	4478,1	1133,6	60	0,0	1
gb8-96	12	1355,7	25,4	0,4305	6771,4	1355,7	437	0,0	1
	15	921,2	320,7	0,4404	7597,3	1822,8	7200	-0,5	1
	12	1865,7	0,0	0,5968	5118,8	1865,7	167	0,0	59
	12	1356,1	26,4	0,4302	6771,4	1356,1	638	0,0	1
	12	1356,1	26,4	0,4302	6771,4	1356,1	150	0,0	1
Moyenne	10,9	1083,1	24,2	0,4052	4800,4	1083,1	118,0	0,0	1,0
	11,9	740,3	246,7	0,4133	5294,4	1448,9	2614,5	-0,1	1,0
	10,8	1459,0	0,0	0,5479	3817,0	1459,0	38,6	0,0	27,4
	11,0	1084,5	30,0	0,4041	4826,5	1084,5	170,8	0,0	1,0
	11,0	1084,5	30,0	0,4041	4826,5	1084,5	45,0	0,0	1,0

TAB. 4.16: Résultats des 3 critères de QoS pour instances de Cordeau modifiée –3

tant en terme de minimisation de la distance totale parcourue, $-24,0\%$ en moyenne, mais a un impact aussi fort ($+27,8\%$) sur le temps cumulé pendant lequel les véhicules se déplacent. Les temps de calcul subissent une forte augmentation ($+2585\%$) explicable par l'augmentation importante de la densité du graphe ($+899\%$) et la faible importance des temps de calcul sur le 1-graphe, souvent inférieurs à la seconde. Dans les temps de calcul, nous ne comptons pas le temps imparti à la construction des graphes.

Nous constatons que les deux méthodes d'optimisation du taux de remplissage convergent assez rapidement. La maximisation du remplissage (donc la minimisation de son inverse qui est inscrite dans les tableaux sous le nom *tx remp.*) produit des solutions très similaires à celles produites par la minimisation de la distance totale parcourue sur un 1-graphe. On peut expliquer ces deux phénomènes par le fait que l'obligation de servir toutes les requêtes borne fortement les valeurs prises par le numérateur et le dénominateur du taux de remplissage qui n'est autre que la distance totale parcourue (dans le cas de la maximisation du taux de remplissage). Nous signalons aussi que malgré leurs différences, les solutions optimales fournies par la méthode directe et la méthode itérative sont strictement identiques pour toutes instances traitées.

Le critère du temps perdu engendre des solutions avec plus de distance parcourue pour un gain en temps perdu relativement faible, surtout pour les instances de type 'c'. Cependant, il est remarquable que même avec un nombre de véhicules limité, la durée roulée cumulée est assez proche de celle obtenue pour les autres critères.

Ces résultats obtenus sur des instances académiques ne suffisent pas à juger définitivement l'optimisation suivant ces différents critères. Des expériences sur des instances réelles doivent compléter cette analyse.

Troisième partie

Étude de cas : Le TAD du Pays du Doubs Central

Chapitre 5

L'application développée

RÉSUMÉ : *L'application produite pour le TAD du Doubs Central est le sujet de ce chapitre. Après un bref aperçu des besoins et de la solution proposée (section 5.1), nous présentons la partie d'optimisation interne à l'application, dans la section 5.2. Le noyau de calcul des tournées est une heuristique d'insertion. Nous comparons ses performances avec une adaptation de la méthode de génération de colonnes du chapitre 4.*

5.1 TAD à mettre en place

5.1.1 Description générale du TAD à réaliser

Le Pays du Doubs Central, regroupant quelques 25000 habitants répartis dans 99 communes dont 5 bourgs et associant 4 Communautés de Communes, est doté depuis janvier 2006 du TAD TADOU. La carte de la figure 5.1 montre la répartition des densités de population et la structure des flux domicile-travail alternants dans le pays. L'objectif, dans ce territoire rural à faible densité de population, consiste à regrouper des voyageurs dans des véhicules parcourant des trajets optimisés en fonction des demandes. L'équipe pluridisciplinaire, comprenant des géographes, des informaticiens, des ingénieurs en transport et des élus locaux souhaite activer deux leviers complémentaires pour favoriser le regroupement :

- par des réductions de coût octroyées aux clients qui souhaitent voyager ensemble, quitte à leur faire supporter quelques détours et pertes de temps raisonnables,
- par l'utilisation de méthodes d'optimisation de tournées visant à minimiser les coûts et, par conséquent, réduire les détours et les pertes de temps pour les passagers.

Pour le Doubs Central, les tournées des véhicules (horaires de passage et parcours) sont définies à partir des demandes de transport des usagers. Les véhicules utilisés (en sous-traitance) sont ceux des compagnies de taxis opérant sur le territoire du Doubs

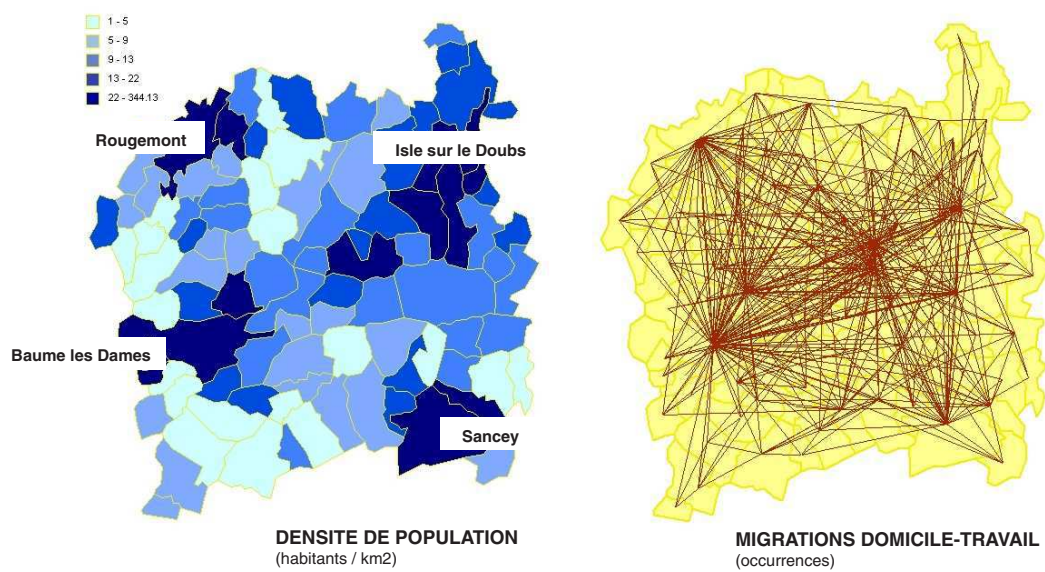


FIG. 5.1: Densités de populations et migrations alternantes dans le Pays du Doubs Central (Josselin, 2005)

Central, soit une dizaine de véhicules potentiellement. Ce service étant fortement subventionné par les collectivités, son coût de fonctionnement doit être minimal. C'est avec cet objectif que nous nous proposons de calculer, une fois les demandes de transport recensées, des itinéraires de taxis optimisés en terme de coût pour la collectivité.

Les tournées des taxis sont calculées jour après jour, pour une journée entière. Un système centralisé de réservation accueille jusqu'à la veille au soir les requêtes des utilisateurs. Chaque usager propose un horaire de prise en charge ou d'arrivée plus ou moins précis parmi un semis de 170 arrêts, le chauffeur ayant ensuite la liberté d'ajuster le lieu d'arrivée. En fonction des détours maximaux possibles, on lui indique à la réservation l'horaire théorique d'arrivée ou de prise en charge le cas échéant (selon une fourchette dont la borne supérieure constitue la limite maximale à ne pas dépasser par le service). Le calcul des tournées est lancé une fois les réservations bloquées pour le lendemain. Les usagers sont ensuite rappelés afin de leur spécifier leur heure effective de prise en charge. Enfin, les feuilles de route correspondantes sont communiquées aux taxis. Ces derniers restent libres de s'échanger les courses par la suite. Un taxi ne peut pas regrouper ses propres clients avec ceux du TAD, à cause des risques de non conformité aux engagements vis-à-vis du client. Le système est de plus défini par :

- le niveau de demandes de mobilité (lié à la structure des semis de populations sur le territoire),
- le nombre de véhicules disponibles (non extensible),
- le nombre et la position de dépôts de ces véhicules (les adresses des compagnies de taxis généralement),
- le calcul du coût préconisé pour rémunérer les compagnies de taxis (prise en compte du haut-le-pied ¹, fixation du facteur kilométrique et du montant de la prise en charge),
- quatre forfaits tarifaires (selon la zone et le regroupement),
- les détours autorisés (valeurs temporelles maximales à ne pas dépasser par trajet).

Le TAD du Doubs Central correspond à un problème de « Dial-a-Ride » réalisé par une flotte de plusieurs véhicules, utilisant des données statiques, sous contrainte de fenêtres de temps. Les caractéristiques principales de ce DARP sont les suivantes :

1. plusieurs véhicules de capacité limitée, basés dans des dépôts,
2. plusieurs requêtes de transport origine-destination émanant de plusieurs personnes, soumises à des contraintes de qualité de service (respect des fenêtres de temps et des délais),
3. matrices des temps, distances et/ou coûts, correspondant au réseau de transport,
4. objectif de minimisation des coûts,
5. toutes ces données sont connues à l'avance.

5.1.2 Définition du problème de calcul de tournées

La collectivité est le client de l'application, son objectif est que toutes les demandes des clients soient satisfaites à un moindre coût pour la collectivité.

¹trajet à vide du taxi pour aller chercher un client.

5.1.2.1 Description des données

Le service doit répondre à R requêtes de transport d'un point de ramassage (élément de l'ensemble \mathcal{R}^+) vers un point de livraison (élément de l'ensemble \mathcal{R}^-). Servir une requête consiste à ramasser un nombre F_r de personnes en $+r$ et à toutes les déposer en $-r$; on pose $F_{+r} = -F_{-r} = F_r$. Ainsi, des personnes partant du même endroit pour des destinations différentes sont modélisées par des requêtes différentes. On dispose d'une flotte \mathcal{K} de K véhicules, de dépôts, de capacités, de coûts fixes d'utilisation journalière et de coûts kilométriques respectifs o^k, F^k, pc^k et Ckm^k . On regroupe sous le nom de service les ramassages, les livraisons ainsi que les départs et les arrivées aux dépôts. Un arc $(v_i, v_j)^e$ est un chemin (dans le réseau routier) reliant le service i au service j . On définit ainsi un p -graphe complet orienté $G = (\mathcal{V}, \mathcal{A})$ doublement valué, où \mathcal{V} est l'ensemble des sommets v_i représentant les différents services ($\mathcal{V} = \{v_i : i \in \mathcal{R}^+ \cup \mathcal{R}^-\} \cup \{v_{o^k} : k \in \mathcal{K}\}$). Les arcs $(v_i, v_j)^e$ sont pondérés par les valeurs D_{ije} et T_{ije} . Pour chaque arc $(v_i, v_j)^e$ et véhicule k on associe un coût C_{ije}^k fonction de la distance, $C_{ije}^k = Ckm^k \times D_{ije}$. L'arc $(v_i, v_j)^0$ représente le plus court chemin en temps de i vers j . Le service i doit démarrer dans la fenêtre de temps $[T_i^{inf}, T_i^{sup}]$. Comme les formules (5.1)-(5.3) le montrent, ces fenêtres de temps sont déduites de la requête client (*i.e.*, la date limite d'arrivée, notée $T_{r^-}^{sup}$) et du délai maximal autorisé, pourcentage du temps de parcours du chemin le plus court, et noté $delai_r$. On se ramène ainsi à un problème de type m -PDPTW standard.

$$T_{r^+}^{sup} = T_{r^-}^{sup} - T_{r^+r^-0} \quad (5.1)$$

$$T_{r^+}^{inf} = T_{r^-}^{sup} - delai \quad (5.2)$$

$$T_{r^-}^{inf} = T_{r^+}^{inf} + T_{r^+r^-0} \quad (5.3)$$

Les véhicules partent à vide, $F_{o^k} = 0$.

Pour déterminer si la tournée du véhicule k passe par l'arc $(v_i, v_j)^e \in \mathcal{A}$, on utilise la variable de décision δ_{ije}^k dont la valeur est fixée à 1 si l'arc est emprunté et 0 sinon. La variable t_i^k représente l'instant où le véhicule k débute le service en $v_i \in \mathcal{V}$ et f_i^k représente la charge (nombre de passagers présents dans le véhicule k) à la fin du service en $v_i \in \mathcal{V}$.

5.1.2.2 Modèle mathématique

$$\min \text{Lex} \left(\sum_{k \in \mathcal{K}} \sum_{(v_i, v_j)^e \in \mathcal{A}} \delta_{ije}^k C_{ije}^k(0) + \sum_{k \in \mathcal{K}} \sum_{(v_{o^k}, v_i)^e \in \mathcal{A}} pc_k \delta_{o^k i^e}^k, \sum_{r \in \mathcal{R}} (T_{r^-}^{sup} - t_{r^+}^k) \right) \quad (5.4)$$

s.c.q.

$$\sum_{k \in K} \sum_{(v_{r^+}, v_j)^e \in A} \delta_{r^+ j^e}^k = 1 \quad \forall r \in \mathcal{R}, \quad (5.5)$$

$$\sum_{(v_{o_k}, v_j)^e \in A} \delta_{o_k j^e}^k = 1 \quad \forall k \in K, \quad (5.6)$$

$$\sum_{(v_{r^+}, v_j)^e \in A} \delta_{r^+ j^e}^k - \sum_{(v_j, v_{r^-})^e \in A} \delta_{j r^-}^k = 0 \quad \forall k \in K, \forall r \in \mathcal{R}, \quad (5.7)$$

$$\sum_{(v_i, v_j)^e \in A} \delta_{ij^e}^k - \sum_{(v_j, v_i)^e \in A} \delta_{ji^e}^k = 0 \quad \forall k \in K, \forall j \in V, \quad (5.8)$$

$$\delta_{ij^e}^k \left(t_i^k + s_i + T_{ij^e}^k - t_j^k \right) = 0 \quad \forall k \in K, (v_i, v_j)^e \in A \setminus \{\mathcal{R}^- \times \mathcal{R}^+\}, \quad (5.9)$$

$$\delta_{ij^e}^k \left(t_i^k + s_i + T_{ij^e}^k - t_j^k \right) \leq 0 \quad \forall k \in K, (v_i, v_j)^e \in A \cap \{\mathcal{R}^- \times \mathcal{R}^+\}, \quad (5.10)$$

$$T_i^{inf} \leq t_i^k \leq T_i^{sup} \quad \forall k \in K, i \in V, \quad (5.11)$$

$$t_{r^+}^k + s_{r^+} + T_{r^+ r^-}^k \leq t_{r^-}^k \quad \forall k \in K, r \in \mathcal{R}, \quad (5.12)$$

$$\delta_{ij^e}^k \left(f_i^k + F_{ij^e} - f_j^k \right) = 0 \quad \forall k \in K, (v_i, v_j)^e \in A, \quad (5.13)$$

$$f_{r^+}^k \sum_{(v_{r^+}, v_j)^e \in A} \delta_{r^+ j^e}^k \leq F^k \quad \forall k \in K, r \in \mathcal{R}, \quad (5.14)$$

$$f_{r^-}^k \sum_{(v_i, v_{r^-})^e \in A} \delta_{i r^-}^k \leq F^k + F_{i r^-}^e \quad \forall k \in K, r \in \mathcal{R}, \quad (5.15)$$

$$\delta_{ij^e}^k \in \{0, 1\} \quad \forall k \in K, (v_i, v_j)^e \in A. \quad (5.16)$$

avec $\delta_{o_k r^-}^k = \delta_{r^+ m_k}^k = 0$; $\delta_{i o_k}^k = 0 \forall i \neq m_k$; $\delta_{m_k i}^k = 0 \forall i \neq o_k$; $L_{o_k} = L_{m_k} = d_{m_k o_k}^e(0) = d_{m_k o_k}^e(1) = d_{i m_k}^e(1) = d_{o_k i}^e(1) = 0$; $A_{o_k} = A_{m_k} = -\infty$ et $B_{o_k} = B_{m_k} = \infty$.

5.1.2.2.1 Explication des contraintes

[5.4] La fonction objectif minimisant le coût total des trajets et les coûts d'utilisation des véhicules. Dans un deuxième temps, le temps perdu est minimisé.

[5.5] Chaque requête est servie exactement une fois et ceci par un seul véhicule.

[5.5] Exactement un véhicule k quitte le nœud de ramassage r^+ et ce une seule fois.

[5.6] Un véhicule effectue au plus une tournée. Le véhicule k quitte exactement une fois son dépôt de départ; on considère qu'il n'effectue pas de tournée, s'il va directement à son dépôt d'arrivée.

[5.7] Une requête est servie par un seul véhicule. Si le véhicule k quitte le nœud de ramassage r^+ alors k arrive au nœud de livraison r^- correspondant.

[5.8] Le véhicule k quitte un nœud v_j autant de fois qu'il le rejoint; que j soit un ramassage, de livraison ou de dépôt.

[5.9] et [5.10] Si le véhicule k emprunte l'arc de v_i vers v_j , il doit débiter le service en j après avoir pu finir le service en i puis voyagé par $(v_i, v_j)^e$. Un temps de latence (ou d'attente) est toléré uniquement après une livraison suivie d'un ra-

massage. Ces contraintes assurent l'élimination des sous-tours² (Desrosiers et al., 1984).

[5.11] Le véhicule k respecte la fenêtre de temps pour le début du service au nœud v_i .

[5.12] Le ramassage doit se faire avant la livraison.³

[5.13] Les véhicules ont l'obligation d'effectuer la requête complète (on prend ou on dépose toutes les personnes) au passage en v_j . On admet qu'on ne dépose personne en route.

[5.14] Le nombre de personnes dans le véhicule k après un ramassage en v_i est compris entre le nombre de personnes à prendre en v_i et la capacité de k .

[5.15] Après une livraison en v_{-i} , il doit y avoir au moins autant de places libres dans le véhicule que ce qui doit être chargé en v_i .

[5.16] Le véhicule k prend ou ne prend pas l'arc numéro e de v_i vers v_j .

La majeure différence entre ce modèle et celui étudié dans le chapitre 4 est la présence de différents véhicules. Les contraintes sur les temps d'attente absurdes n'interviennent qu'au deuxième niveau de l'optimisation, lors de la minimisation des temps perdus. Ce modèle mathématique peut être aisément linéarisé. Résoudre ce problème en utilisant sa relaxation linéaire est possible en y ajoutant un certain nombre de coupes non triviales (Ropke et al., 2007). Nous donnons ici le modèle à titre informatif puisque nous avons opté dans cette thèse pour une résolution du modèle d'affectation généralisé par génération de colonnes.

5.1.3 L'application TADOU

L'application TADOU (figure 5.2) développée par l'équipe de recherche et maintenant propriété de la société Prorentsoft permet à l'opérateur d'enregistrer les requêtes de transport, puis de créer manuellement ou automatiquement des tournées. Un module permet aussi la modification manuelle des tournées puis le contrôle automatique de leur validité. Enfin les tournées peuvent être éditées afin d'être transmises aux compagnies de taxis. Sur la figure 5.3 on voit comment l'opérateur enregistre une nouvelle requête et peut entre autres lui attribuer des fenêtres de temps plus ou moins souples.

Les données relatives au système de transport (véhicules, réseau routier, arrêts...) et aux requêtes sont stockées dans une base de données Microsoft Access. Le noyau de calcul des tournées est en C++. L'interface avec la base de données et quelques prétraitements sont codés en Microsoft Visual Basic.

²un sous-tour est un circuit n'ayant pas de dépôt pour origine.

³à remplacer par $t_{ik} \leq t_{i-k} \forall k \in \mathcal{K}, i \in \mathcal{R}^+$ si l'inégalité triangulaire n'est pas respectée par les T_{ij} .



FIG. 5.2: Accueil de TADOU

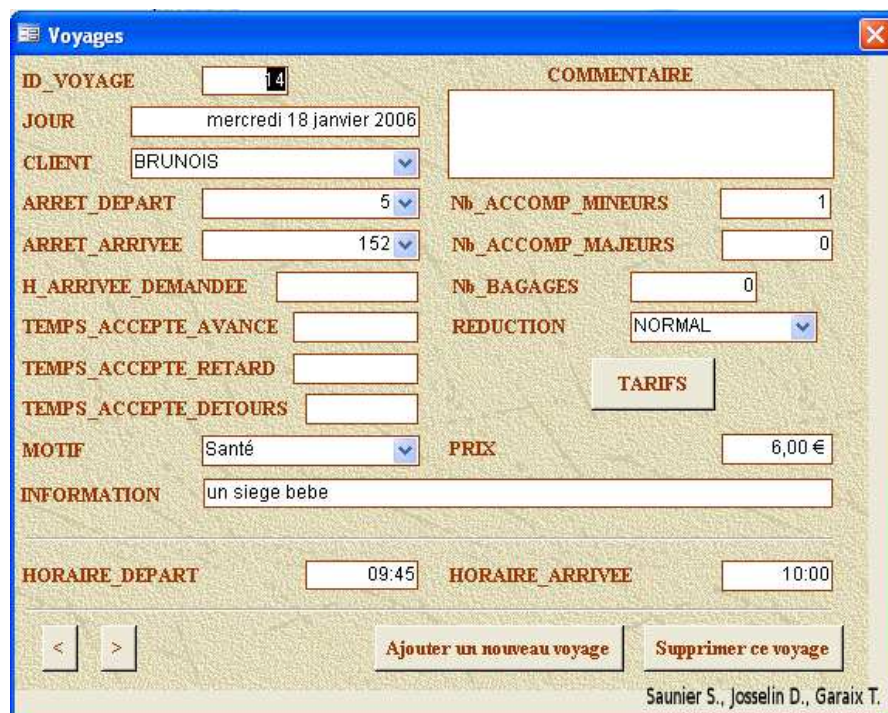


FIG. 5.3: Enregistrement d'une requête avec paramétrage de son niveau de qualité de service

5.2 Heuristique d'optimisation

5.2.1 Résolution par un algorithme d'insertion

Le volume de données à traiter dans le Doubs Central permet d'envisager une approche exacte. Cependant, deux contraintes nous ont orientés vers une autre approche. Premièrement le projet du Doubs Central est un projet d'action recherche qui par définition n'offre pas un système de transport clairement défini à optimiser. En effet de nombreuses variantes mineures ou majeures ont été considérées au cours d'un développement qui devait être opérationnellement intégré avant la fin de la première année de thèse. Certaines de ces modifications auraient pu être rédhibitoires ou auraient entraîné de lourds ajustements pour une méthode complexe. De plus, la société assurant à présent la maintenance et le développement du produit possède un noyau d'optimisation basé sur un algorithme dont la maîtrise nécessite peu de connaissances, ce qui était un de leurs souhaits en vue de s'attaquer au marché des TAD en temps réels. Nous avons donc proposé un algorithme (codé en C++) standard d'insertion combiné à un opérateur simplifié de recherche locale. Cette approche nous semble raisonnablement répondre aux attentes des divers acteurs du projet et nous a offert un certain confort au niveau du temps de réalisation par rapport aux délais de livraison du premier prototype opérationnel.

5.2.1.1 Idée générale

Nous proposons un algorithme d'optimisation décrit par la figure 5.4 qui fonctionne en trois phases :

- une première phase pendant laquelle on construit des tournées réalisables par une procédure d'insertion gloutonne (algorithme 5),
- une seconde phase où, par le biais d'insertions et de suppressions de requêtes dans les tournées, on cherche à améliorer la qualité de la solution,
- une dernière procédure fixe les dates de réalisations des services.

L'insertion des requêtes se fait dans la tournée où elle est la moins coûteuse. L'insertion d'une requête dans une tournée ne doit pas modifier l'ordre déjà établi entre les sommets composant cette tournée à une itération précédente.

5.2.1.2 Procédure d'insertion gloutonne

Des séquences vides sont associées à chaque véhicule. Les requêtes sont insérées une à une dans un ordre prédéfini. Une nouvelle insertion ne remet pas en cause l'ordre établi par les insertions précédentes. Le choix des arcs sélectionnés est en revanche ré-optimisé à chaque insertion. En effet, l'insertion d'une requête peut contraindre à l'utilisation de chemins plus rapides mais plus coûteux. Le calcul d'une tournée optimale – position d'insertion et sélection des arcs – est un cas particulier du problème de Sélection des Arcs dans une Séquence Fixée (PSASF, voir section 3.3.3) avec trois ressources,

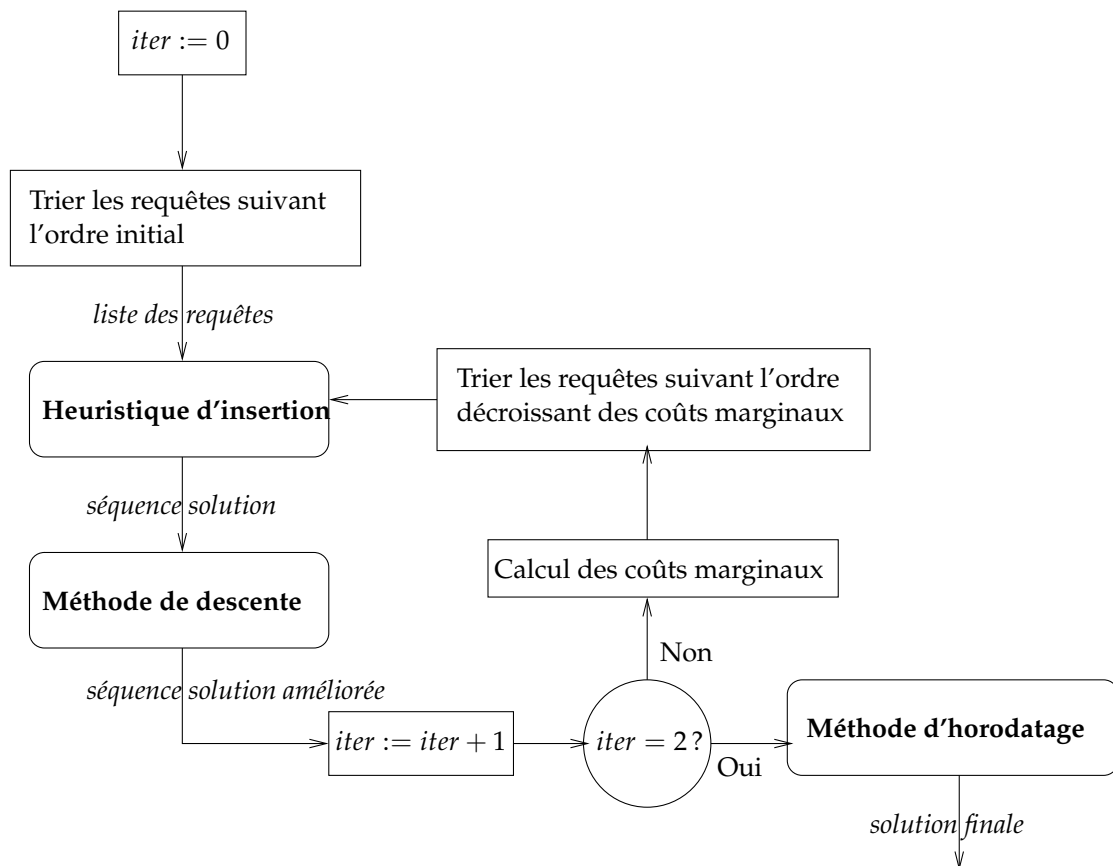


FIG. 5.4: Résolution en trois phases

une pour la durée (T), une pour les flux de passagers (F) et une pour représenter l'insertion de la nouvelle requête r (I), dans un graphe adapté. Cette dernière ressource prend la valeur 1 lorsque la nouvelle requête est en cours, la valeur 2 lorsque sa livraison a été effectuée et 0 lorsque la nouvelle requête n'est pas commencée. L'insertion est bien sûr valide uniquement si la ressource I est à la valeur 2 à l'arrivée au dépôt. Notons que si la quantité de passagers maximale rencontrée dans la séquence initiale S_ω est inférieure à la différence entre la capacité du véhicule (F^k) et le nombre de passagers de la nouvelle requête, alors la ressource sur le flux de passagers peut être ignorée sans risque de perte d'optimalité. Afin d'appliquer notre algorithme, nous construisons le p -graphe acyclique $G_{r\omega} = (V_{r\omega}, A_{r\omega})$ dont les sommets sont ceux de la séquence initiale S_ω auxquels nous ajoutons un sommet pour chaque position d'insertion envisageable pour r^+ et r^- , formant ainsi les ensembles respectifs $Clone(r^+)$ et $Clone(r^-)$. Nous précisons que déterminer si une position d'insertion est envisageable ou non se fait ici par une simple vérification sur les fenêtres de temps et de flux. Tous les arcs respectant l'ordre imposé par $S_\omega =$ sont inclus dans $A_{r\omega}$, comme l'indique la figure 5.5. Un nouvel indicage des sommets est établi (à côté des sommets sur la figure). Le parcours respectant l'ordre de ces indices assure de ne jamais passer deux fois par un même sommet, ce qui évite de contrôler l'élémentarité des chemins partiels pour les sommets de S_ω .

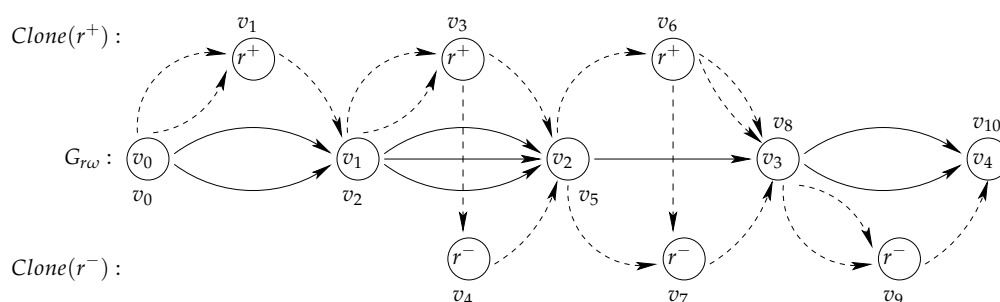


FIG. 5.5: Insertion d'une requête dans une séquence – construction du p -graphe

L'algorithme 5 accompagné des deux fonctions 6 et 7, est une adaptation de l'algorithme 2 de la section 3.3.3 résolvant (PSASF).

Cet algorithme s'applique au cas du graphe simple (ne représentant pas tous les chemins non-dominés entre chaque paire de sommets) à condition de modifier, dans la fonction de dominance, l'inégalité entre les niveaux de consommation de la ressource I par une égalité. En effet, dans ce cas, il n'est pas sûr de pouvoir rencontrer un arc entre deux sommets i et j de S_ω dominant (en fonction du coût et de la durée) un détour entre i et j passant par r^+ et/ou r^- . Ainsi, on ne peut pas décider de la dominance entre deux chemins partiels qui n'ont pas le même niveau d'avancement dans le service de la nouvelle requête r .

Il paraît raisonnable de supposer que des bonnes tournées (peu coûteuses) vont effectuer des regroupements de requêtes intéressants. De tels regroupements doivent répondre à des contraintes de proximité spatiale et temporelle. Il est clair que le regroupement de deux requêtes « proches » spatialement et temporellement, a des chances d'améliorer le coût de la solution. L'insertion d'une requête s'effectuant au moindre

Algorithme 5 : Procédure d'insertion

Données : G_{rw} une séquence
Résultat : $CH_{-(R+k)}$ les chemins arrivant au dépôt

- 1 Initialisation : $CH_{R+k} :=$ un chemin au départ du dépôt;
- 2 $i := R + k$;
- 3 **tant que** $i \neq -(R + k)$ **faire**
- 4 **pour chaque** chemin partiel $ch \in CH_i$ **faire**
- 5 **pour chaque** arc $(v_i, v_i)^e$ **faire**
- 6 $ch' := extension(ch, (v_i, v_i)^e)$;
- 7 **si** $ch' \neq \emptyset$ **alors**
- 8 **si** $dominance(ch') = faux$ **alors**
- 9 | $CH_i := CH_i \cup \{ch'\}$;

Algorithme 6 : Fonction d'extension

Données : $ch \in CH_i, (v_i, v_j)^e \in A_w$
Résultat : \emptyset ou ch' extension de ch

- 1 **si** $j \in Clone(r^+), I_{ch} = 0$ **alors**
- 2 | $I_{ch} := 1$;
- 3 **sinon**
- 4 **si** $j \in Clone(r^-), I_{ch} = 1$ **alors**
- 5 | $I_{ch} := 2$;
- 6 **sinon**
- 7 **si** $j \notin V_w$ ou $(j = -R - k$ et $I_{ch} \neq 2)$ **alors**
- 8 | retourne \emptyset ;
- 9 $T_{ch'} := \max(T_{ch} + T_{ij^e}, T_j^{inf})$;
- 10 $F_{ch'} := F_{ch} + F_{ij^e}$;
- 11 $C_{ch'} := C_{ch} + C_{ij^e}$;
- 12 **si** $T_{ch'} > T_j^{sup}$ ou $F_{ch'} > F^k$ **alors**
- 13 | retourne \emptyset ;
- 14 **sinon**
- 15 | retourne ch' ;

Algorithme 7 : Fonction de dominance

Données : $ch \in CH_i$

Résultat : ch a un dominant ?

```

1 pour chaque chemin partiel  $ch' \in CH_i$  faire
2   si  $T_{ch} \leq T_{ch'}$  et  $C_{ch} \leq C_{ch'}$  et  $F_{ch} \leq F_{ch'}$  et  $I_{ch} \leq I_{ch'}$  alors
3      $CH_i := CH_i \setminus \{ch'\}$ ;
4   sinon
5     si  $T_{ch'} \leq T_{ch}$  et  $C_{ch'} \leq C_{ch}$  et  $F_{ch'} \leq F_{ch}$  et  $I_{ch'} \leq I_{ch}$  alors
6       retourne vrai;
7 retourne faux;

```

coût, elle a tendance à privilégier le critère spatial dans l'optimisation. Nous proposons donc de trier initialement les requêtes suivant une règle de priorité temporelle basée sur la date de livraison souhaitée.

Une autre technique est de déterminer l'ordre de traitement des requêtes à partir d'une solution obtenue par l'algorithme à partir d'un ordre quelconque. Un bénéfice est associé à la suppression momentanée de chaque requête. Le calcul de ce bénéfice induit la résolution d'un nouveau PSASF sans, bien sûr, la ressource représentant l'insertion de la requête. Une fois tous les bénéfices de suppressions calculés, la procédure d'insertion est relancée sur les requêtes dans l'ordre des valeurs décroissantes de ces bénéfices.

5.2.1.3 Optimisation locale par suppression puis insertion

L'idée est de retirer certaines requêtes de la solution, puis de les réinsérer au moindre coût (avec la même procédure d'insertion que pour l'algorithme 5) d'insertion gloutonne. La nouvelle solution est ainsi soit identique à l'ancienne, soit plus avantageuse. On peut espérer modifier la solution courante par cette opération puisque, par rapport à l'insertion lors de la première exécution de l'algorithme 5, toutes les requêtes sont desservies. Dans un premier temps, ce traitement est appliqué aux requêtes dans leur ordre initial d'insertion (qui est l'ordre s'étant révélé comme le plus efficace expérimentalement). Ensuite, le choix de la requête à insérer se fait aléatoirement jusqu'à atteindre une limite de temps de calcul imposée ou que plus aucune amélioration de la solution ne puisse être espérée.

5.2.1.4 Horodatage

À la fin de la seconde étape, les séquences et les arcs correspondant aux tournées sont choisis. Le premier niveau de la fonction objectif est donc atteint sur l'ensemble des séquences obtenues. Il s'agit à présent de résoudre le problème d'horodatage de minimisation du temps perdu, dans un 1-graphe, présenté dans la section 3.3.2. Deux

procédures récursives, la première parcourant la tournée du dépôt d'arrivée vers celui de départ et la seconde parcourant la tournée en sens inverse, permettent de construire un horodatage fixant les ramassages au plus tard, afin de minimiser le temps perdu, et les livraisons au plus tôt, afin d'éviter les temps d'attente absurdes (voir les sections 3.2 et 3.3.2 pour plus de détails).

5.2.2 Résultats

Afin d'évaluer les performances de notre heuristique, nous comparons les résultats obtenus par l'heuristique avec ceux obtenus par génération de colonnes. Le problème maître doit être légèrement modifié pour permettre de considérer une flotte hétérogène de véhicules, composée de M véhicules type représentant des véhicules aux mêmes caractéristiques. À chaque itération, les problèmes esclaves inhérents à chaque véhicule type sont résolus dans un ordre aléatoire (biaisé par la valeur de la variable duale associée). La recherche de tournées est stoppée dès qu'un nombre maximal de tournées de coût réduit négatif est atteint. Le nombre maximal de tournées à remonter au problème maître est décrémenté en fonction du temps déjà imparti à la résolution des problèmes esclaves d'une même itération. Le problème maître présenté dans la section 4.1.1 prend la forme suivante :

$$(MP) \quad \min \sum_{\omega \in \Omega} c_{\omega} \lambda_{\omega} \quad (5.17)$$

s.c.q.

$$\sum_{\omega \in \Omega} \rho_{\omega}^r \lambda_{\omega} = 1 \quad \forall r = 1, \dots, R, \quad [\pi_{\rho}^r] \quad (5.18)$$

$$\sum_{\omega \in \Omega} \lambda_{\omega} \leq K^m \quad \forall m = 1, \dots, M, \quad [\pi_{\kappa}^m] \quad (5.19)$$

$$\lambda_{\omega} \in \{0, 1\} \quad \forall \omega \in \Omega. \quad (5.20)$$

Le problème esclave n'est pas modifié par ces nouvelles contraintes puisqu'elles sont indépendantes d'un véhicule à l'autre. Nous présentons des résultats sur huit instances réalistes (le tableau 5.1) dont la construction est expliquée dans la section 6.1.2, et sur les instances modifiées de Li et Lim (les tableaux 5.2, 5.3 et 5.4). Le temps de calcul de l'heuristique – en fait de la phase d'insertion/suppression car la construction est très rapide – est limité à 6 secondes. Le temps de calcul de la génération de colonne (colonne *sec. opt.*) inclut 18 secondes consacrées à trois exécutions de l'heuristique pour initialiser l'algorithme.

La méthode heuristique proposée ne parvient que rarement à l'optimum et dans le cas de la recherche dans un p -graphe les solutions obtenues peuvent être jusqu'à 20% supérieures à l'optimum. Néanmoins, il est intéressant de remarquer que ces solutions obtenues rapidement sur le p -graphe (instances '*mr*' et '*mc*'), ont un coût souvent inférieur à celui des solutions optimales calculées sur le 1-graphe (instances '*gr*' et '*gc*').

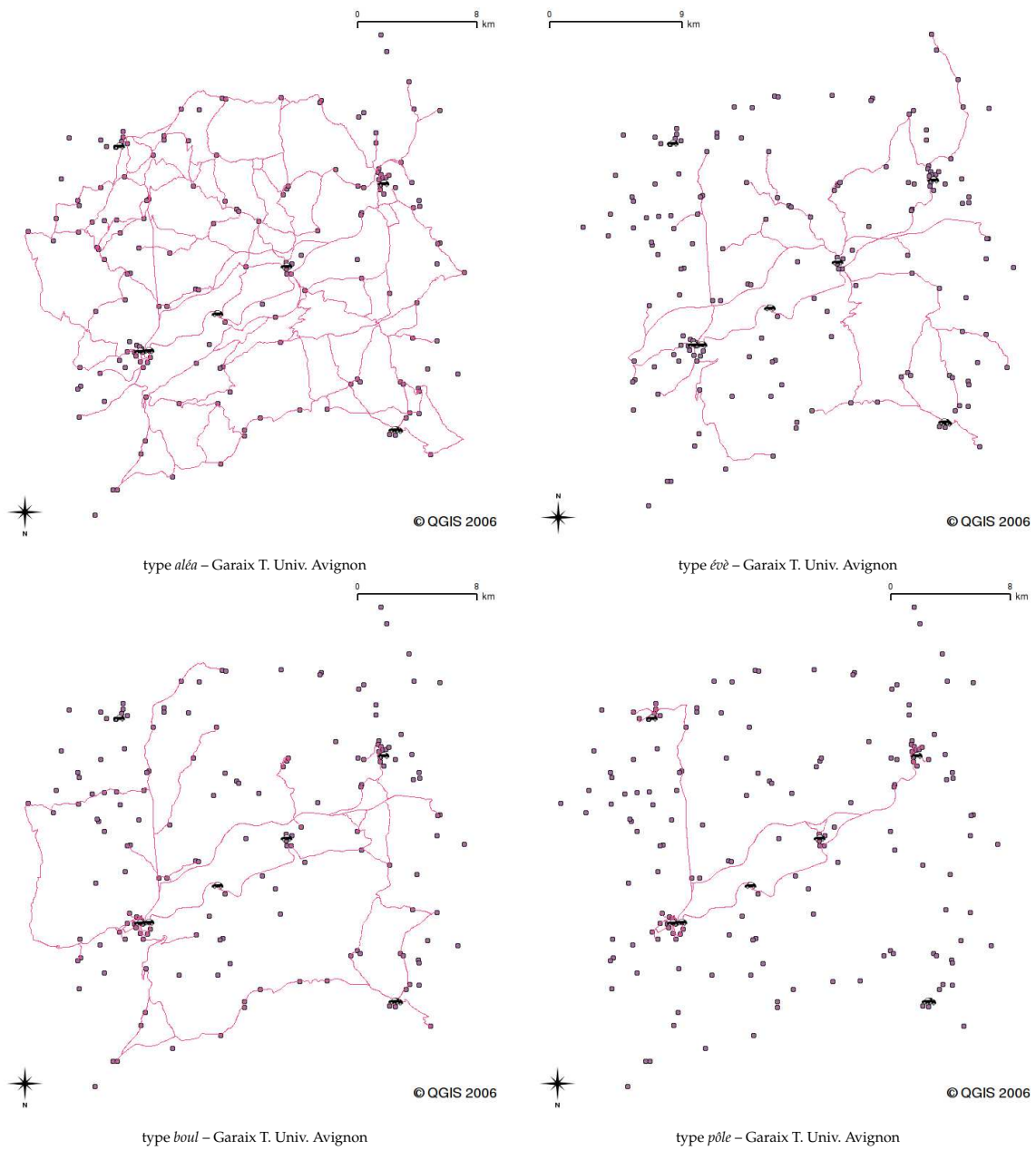


FIG. 5.6: Quatre solutions typées pour des instances à 50 requêtes dans le Doubs Central

nom	veh.	dist.	dist. opt.	%opt	sec. heu.	sec. opt.
mdc199	11	1589,1	1463,6	8,6	6	1144
dc199	11	1747,9	1592,1	9,8	6	60
mdc200	16	1274,5	1273,3	0,1	6	193
dc200	10	1427,0	1425,0	0,1	6	46
mdc201	14	1192,4	1173,2	1,6	6	1742
dc201	13	1286,4	1253,2	2,6	6	234
mdc202	11	1381,6	1155,0	19,6	6	3600
dc202	13	1475,4	1350,5	9,3	6	48
mdc374	11	1570,2	1471,4	6,7	6	693
dc374	12	1710,5	1596,2	7,2	6	36
mdc375	16	1324,2	1323,0	0,1	6	215
dc375	16	1470,5	1467,1	0,2	6	24
mdc376	14	1368,6	1361,3	0,5	6	158
dc376	16	1465,3	1457,5	0,5	6	23
mdc377	14	1453,2	1278,0	13,7	6	2850
dc377	16	1583,2	1494,9	5,9	6	32

TAB. 5.1: Résultats pour huit instances réalistes

nom	veh.	dist.	dist. opt.	%opt.	sec. heu.	sec. opt.
mr101	25	1536,4	1530,3	0,4	6	19,82
gr101	24	2030,6	1988,5	2,1	6	18,12
mr102	23	1467,1	1451,6	1,1	6	23,87
gr102	22	1970,2	1927,2	2,2	6	18,16
mr103	17	1461,0	1398,4	4,5	6	23,36
gr103	17	1888,3	1812,9	4,2	6	18,24
mr104	25	1442,4	1440,6	0,1	6	19,19
gr104	25	1908,7	1897,6	0,6	6	18,15
mr105	25	1570,0	1562,8	0,5	6	20,23
gr105	25	2082,1	2046,4	1,7	6	18,14
mr106	20	1448,5	1426,3	1,6	6	20,39
gr106	18	1873,4	1864,0	0,5	6	18,18
mr107	19	1464,5	1412,5	3,7	6	25,14
gr107	17	1882,9	1834,5	2,6	6	18,17
mr108	27	1592,7	1588,5	0,3	6	18,58
gr108	27	2131,2	2125,7	0,3	6	18,12
mr109	23	1489,8	1451,5	2,6	6	19,06
gr109	23	1948,5	1917,8	1,6	6	18,14
mr110	25	1500,8	1476,7	1,6	6	18,85
gr110	25	1972,6	1961,4	0,6	6	18,1
mr111	22	1427,6	1413,0	1,0	6	21,59
gr111	21	1877,0	1844,0	1,8	6	18,12
mr112	27	1715,4	1705,3	0,6	6	18,21
gr112	27	2187,2	2182,2	0,2	6	18,1

TAB. 5.2: Résultats de l'heuristique pour les instances 'gr' et 'mr'

L'affichage des tournées solutions sur la carte du Doubs Central (figure 5.2.2) apporte des informations supplémentaires quant à l'utilisation des routes. Les quatre so-

nom	veh.	dist.	dist. opt.	%opt.	sec. heu.	sec. opt.
mc101	9	991,8	795,6	24,7	6	77,47
gc101	9	1247,6	1002,8	24,4	6	18,62
mc102	9	961,2	813,4	18,2	6	89,21
gc102	9	1296,8	1060,9	22,2	6	18,82
mc103	13	1027,7	942,8	9,0	6	53,51
gc103	13	1375,0	1250,4	10,0	6	18,6
mc104	28	1461,1	1438,4	1,6	6	21,98
gc104	28	1984,5	1982,2	0,1	6	18,18
mc105	11	982,9	900,1	9,2	6	54,37
gc105	11	1345,7	1185,4	13,5	6	18,59
mc106	13	1009,4	952,5	6,0	6	52,06
gc106	13	1296,3	1244,7	4,1	6	18,48
mc107	14	1000,0	982,5	1,8	6	56,11
gc107	13	1418,9	1305,9	8,6	6	18,36
mc108	12	994,7	941,8	5,6	6	56,9
gc108	12	1278,7	1191,3	7,3	6	18,39
mc109	17	1137,0	1116,6	1,8	6	34,25
gc109	17	1488,3	1453,2	2,4	6	18,37

TAB. 5.3: Résultats de l'heuristique pour les instances 'gc' et 'mc'

nom	veh.	dist.	dist. opt.	%opt.	sec. heu.	sec. opt.
mrc101	22	1775,7	1769,1	0,4	6	20,43
grc101	22	2301,6	2290,9	0,5	6	18,09
mrc102	21	1655,4	1645,8	0,6	6	20,38
grc102	19	2149,1	2145,7	0,2	6	18,16
mrc103	18	1608,5	1545,0	4,1	6	21,29
grc103	17	2166,8	2055,4	5,4	6	18,2
mrc104	27	1916,1	1898,6	0,9	6	19,74
grc104	27	2637,7	2599,8	1,5	6	18,12
mrc105	25	1869,0	1856,4	0,7	6	18,62
grc105	25	2590,4	2543,2	1,9	6	18,12
mrc106	21	1693,0	1693,0	0,0	6	19,48
grc106	20	2189,2	2179,9	0,4	6	18,11
mrc107	29	1927,9	1926,9	0,1	6	18,66
grc107	29	2632,9	2632,9	0,0	6	18,09
mrc108	30	1965,5	1965,5	0,0	6	18,26
grc108	30	2662,9	2662,9	0,0	6	18,1

TAB. 5.4: Résultats de l'heuristique pour les instances 'grc' et 'mrc'

lutions (une par scénario) proposées indiquent que le scénario d'une augmentation de trafic faisant ressortir l'importance des pôles d'habitation entraîne une utilisation presque exclusive de seulement trois routes.

Chapitre 6

Visualisation des tournées

RÉSUMÉ : *Après avoir mis en place des outils permettant la visualisation et la manipulation des tournées de véhicules dans leur environnement géographique, nous générons des instances réalistes correspondant à différents scénarios de demandes de transport. Nous comparons ensuite les solutions obtenues à partir des trois critères de qualité de service que sont la distance totale parcourue, le temps perdu et le taux de remplissage. Pour finir dans la section 6.2, nous étudions comment évaluer la sinuosité des tournées, vue comme un critère de qualité de service lié à la forme des tournées.*

6.1 Plate-forme de travail couplée à un SIG

Afin de pouvoir analyser les interactions entre le système de transport et son environnement, il est nécessaire d'intégrer les résultats de l'activité du TAD dans des outils d'analyse du territoire et des populations couramment utilisés. Certains chercheurs, dont les géographes, ont un besoin en traitement et analyse de gros volumes de données pour générer des statistiques, mais aussi un besoin de visualisation au plus proche de la réalité géographique pour affiner et critiquer les observations statistiques.

Les Systèmes d'Information Géographique ont cette propriété. Ils permettent de traiter de vastes bases de données, pouvant contenir des objets géométriques complexes, et de visualiser sous différentes formes, généralement de type cartographique, les résultats des requêtes effectuées sur ces données. Les produits commerciaux (Mapinfo, Arcgis, Geoconcept...), relativement onéreux, sont très aboutis dans le traitement et l'affichage des résultats des requêtes mais leur utilisation rend l'interaction directe avec la base de données difficile ; ils respectent des standards qui leur sont propres (par exemple la présentation des objets géométriques avec les formats MIF/MID, ESRI shape...) et ne possèdent pas, dans leur version de base, de modules de calcul de tournées de véhicules (au mieux une méthode de calcul de plus court chemin). Pour ces

raisons et dans la perspective de réaliser un outil universitaire libre d'accès, nous nous sommes intéressés aux logiciels libres (*open source*) du domaine.

6.1.1 Structure de la plateforme Opensource

En 2004, l'offre en logiciels libres dans le domaine des SIG est au début de son développement. L'offre s'est largement étoffée depuis mais l'idée d'un projet global intégrant différents modules reste assez lointaine puisque les projets propres à chaque module ne sont pas encore arrivés à maturité. Seule la structuration des objets géométriques au formalisme du projet Postgis (à l'origine, une extension du projet de SGBD¹ Postgres) semble s'imposer. Beaucoup de fonctions sont déjà développées pour leur manipulation et les conversions bidirectionnelles sont possibles avec la plupart des formats de données standard vectoriels ou non. Nous avons donc nous-mêmes assuré l'interopérabilité entre différents projets, comme l'illustre la figure 6.1 où les flèches en pointillés indiquent les connexions restant à ce jour à accomplir.

La base de données est une base Postgres configurée pour la gestion d'objets géographiques Postgis, capable d'utiliser des milliers de Systèmes Référentiels, et construits à partir d'objets géométriques Postgis. Le SIG Quantum GIS permet la visualisation, la manipulation – encore limitée – et l'édition des données, le croisement avec d'autres données provenant de diverses sources (base de données, fichiers Shape, MIF/MID, Raster, Mapserver. . .) et la création de cartes. D'autres modules sont développés ou en projet, comme le couplage à un système GPS. Un serveur de cartes, Mapserver, basé sur un serveur Http Apache et un serveur Php, permet la diffusion de cartes à travers le réseau internet. À cela, vient s'intégrer le noyau de calcul de tournées de véhicules qui prend les données dans la base et y réinjecte les solutions obtenues. Le noyau développé en C++ (et SQL pour certains traitements) a été adapté pour pouvoir devenir, à moindre effort, un module du projet PgRouting qui intègre déjà les résolutions du problème de plus court chemin et du voyageur de commerce, et qui devrait donner naissance à un module dans QuantumGIS. Ce dernier permettrait alors, depuis son interface, de gérer la génération des données, le calcul des tournées et l'analyse des résultats.

6.1.2 Construction d'instances typées

Comme on a pu s'en apercevoir avec les résultats obtenus sur les instances de la littérature, la structure spatiale de la demande (trois classes sont distinguées : groupées, aléatoires et mixtes) a un impact sur le type de solutions trouvées (nombre de véhicules utilisés, distance parcourue. . .). D'autres caractéristiques générales des demandes méritent d'être observées, comme l'orientation des flux pour la dimension spatiale et la répartition dans le temps et taille des fenêtres de temps associées dans la dimension temporelle. Nous souhaitons aussi étudier des caractéristiques plus liées au détail du terrain comme le taux de fréquentation des routes, la forme des tournées ou la répartition géographique des véhicules. Il est évident que ce type d'observations n'est envisa-

¹SGBD : Système de Gestion de Base de Données.

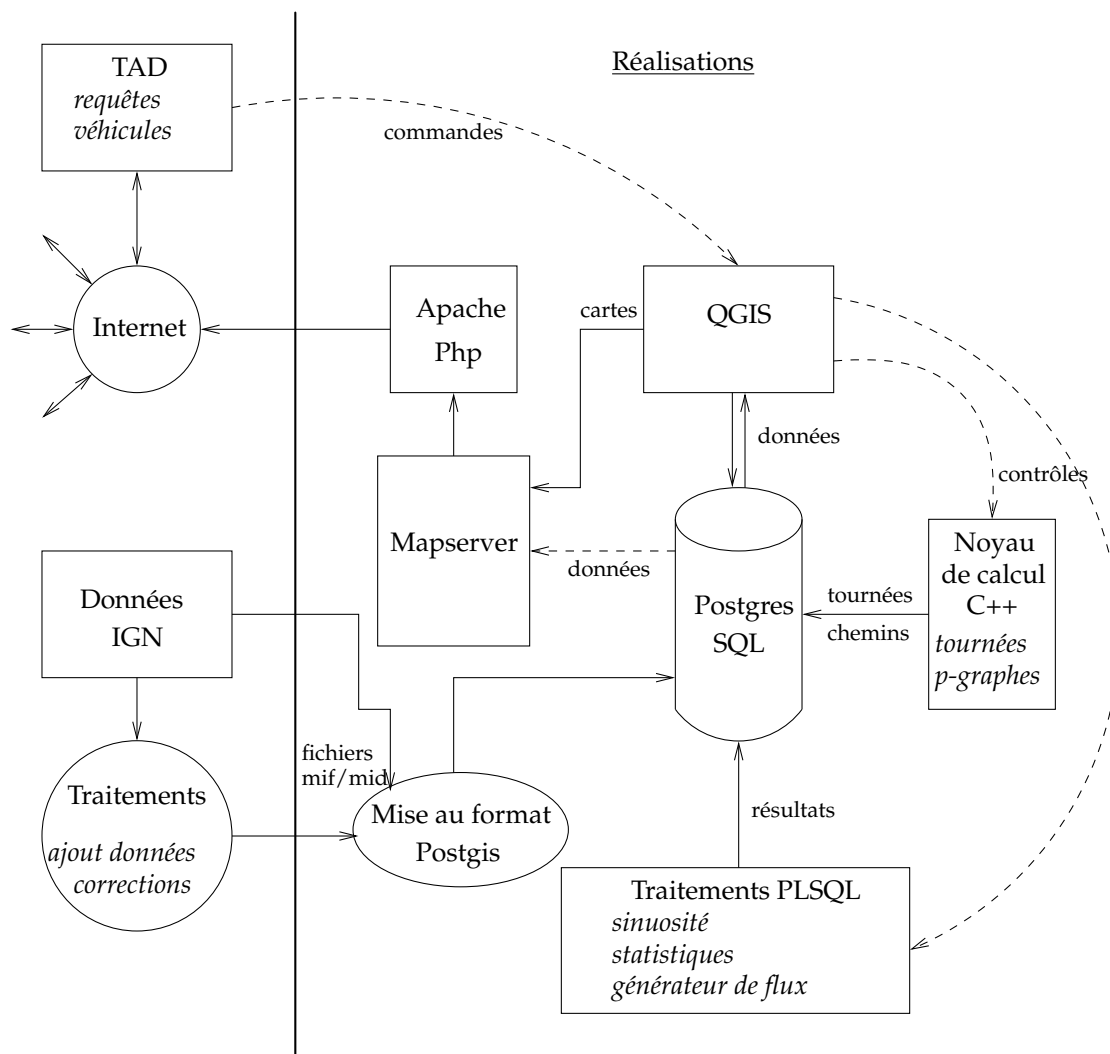


FIG. 6.1: Plate-forme d'édition, de visualisation et de diffusion de tournées de véhicules

geable qu'avec des données réalistes. De plus, on peut s'attendre à ce que l'impact des critères de qualité de services considérés soit très différent sur des données réalistes ; comme nous le laisse imaginer le cas du détour maximal autorisé. Ce dernier est établi en fonction de la distance entre le lieu de ramassage et celui de livraison. Il correspond à une distance euclidienne pour les instances de la littérature et à un itinéraire sur le réseau pour des données réelles. Il est clair que l'itinéraire sur le réseau a des chances de passer par des lieux de ramassage ou de livraison d'autres requêtes ; regrouper ces services dans le même véhicule n'affecterait donc nullement le volume de détour enregistré pour les requêtes concernées. La figure 6.2 présente un exemple où le plus court chemin dans le réseau routier entre le ramassage et la livraison passe exactement par le ramassage d'une autre requête, ce qui est fort possible dans un réseau routier. Pour que ce cas se produise dans le plan euclidien, il faudrait que j^+ se trouve sur le segment $[i^+, i^-]$, ce qui est peu probable.

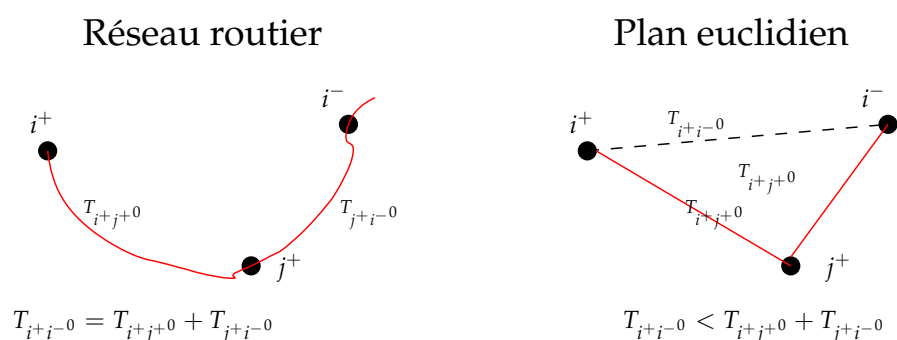


FIG. 6.2: Regroupement sans détours dans un réseau routier

Nous considérons trois axes de différenciation des flux de passagers. Le premier axe est le niveau de polarisation spatiale des flux. Les demandes de transport peuvent être émises tout azimut ou plus ou moins concentrées entre un ou plusieurs pôles dans leur direction et éventuellement leur orientation. Le deuxième axe correspond au niveau de polarisation temporelle des flux ; c'est-à-dire la présence et la répartition des pics de demandes au cours du temps. Le troisième axe représente le niveau de détour autorisé par les passagers.

Nous restreignons notre étude aux cas correspondants à quatre scénarii d'évolution possible du système TADOU. On peut estimer que le fonctionnement actuel qui correspond au premier scénario noté *alea*, traite des demandes équitablement réparties dans l'espace et le temps, puisque le nombre assez faible de demandes journalières (entre dix et trente requêtes par jour) ne permet pas aux concentrations de populations des plus gros bourgs d'avoir un impact significatif sur la structure des flux d'une journée. Le deuxième scénario noté *pôle* correspond à une montée en charge du système qui ferait apparaître les zones les plus densément habitées comme des pôles d'attraction des flux. Deux autres scénarii respectivement notés *boul* et *évèn* peuvent apparaître, ils correspondent à des événements dans une zone géographique restreinte concentrant les flux sur une plage horaire réduite ou bien étendue à toute une journée.

Pour chaque type de flux, nous générons des instances à 50 requêtes car à partir d'un

certain seuil d'occupation des véhicules, la montée en charge n'affecte pas les critères de qualité de service considérés et augmente uniquement le nombre de véhicules utilisés.

Ce travail est rendu possible par le développement d'outils d'intégration des cartes IGN² contenant les informations détaillées du réseau (localisation spatiale en longitude, latitude et altitude, pénibilité, types d'intersection, ponts, péages, nature administrative des routes...), dans la base de données au format Postgis ; puis du développement d'outils paramétrables : un intégrateur et un générateur de flux sur les données réelles.

La carte 6.3 est un aperçu du réseau routier du Doubs Central. Une portion d'autoroute traverse le territoire d'est en ouest en son milieu (trait large). Deux voies rapides (limitées à 110 km/h) traversent le territoire du nord au sud et dans la direction sud-est nord-ouest (lignes traits-points). Le reste du réseau est composé de routes limitées à 70 km/h (en pointillés) et à 50 km/h. Cela représente un graphe de 2 792 tronçons aux caractéristiques identiques eux-mêmes décomposés en 26 585 segments, 2 151 nœuds dont 170 arrêts marqués par des drapeaux, et 8 dépôts représentés par des voitures. Nous avons calculé 166 473 chemins non-dominés (par rapport à la durée et la distance), avec l'algorithme 4 de construction de p -graphe de la section 4.2.1.1.

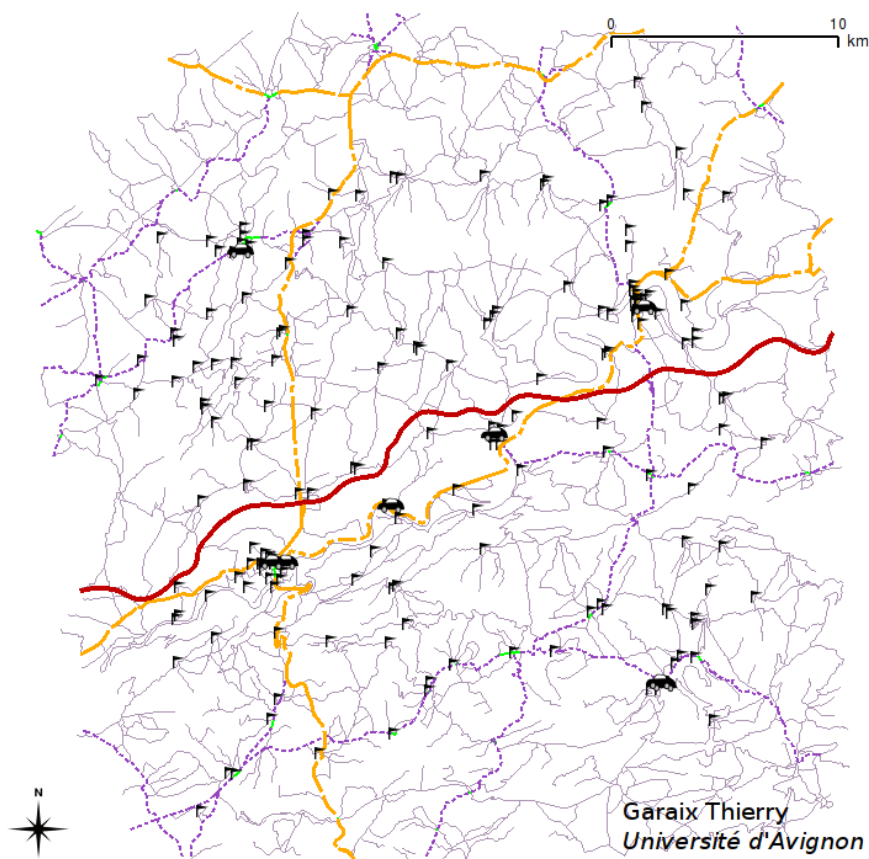


FIG. 6.3: Réseau routier du Doubs Central

²IGN : Institut Géographique National.

Le tableau 6.1 contient les caractéristiques de huit instances générées suivant les scénarii précités et indiqués dans la colonne *modèle*. Les instances sont groupées par

nom	modèle	req.	largeur	écart	dens. 1-g.	dens. p-g.	flux
dc199	alea154	50	8,09	24,27	0,96	4,48	2
dc374	alea154	50	5,39	21,57	0,94	4,40	2
dc200	évèn159	50	5,64	16,93	0,97	2,51	2
dc375	évèn159	50	3,76	15,05	0,96	2,48	2
dc201	boul161	50	6,35	19,06	0,91	3,27	2
dc376	boul161	50	4,24	16,94	0,78	2,87	2
dc202	pôle163	50	7,16	21,48	1,02	6,74	2
dc377	pôle163	50	4,77	19,10	0,99	6,57	2

TAB. 6.1: Caractéristiques de 8 instances réalistes

paires identiques deux à deux. La différence est faite par les contraintes de fenêtres de temps. Les instances dont le nom inclut un nombre compris entre 199 et 202 ont des fenêtres de temps calculées avec des détours (en temps) maximaux égaux à un tiers du plus court chemin en temps entre ramassage et livraison. Les autres instances (numérotées entre 374 et 377) ont des flux identiques pour un même modèle avec un détour maximal autorisé d'un quart du plus court chemin en temps.

6.1.3 Critères de qualité de service et modèles de flux

Les trois critères de qualité de service étudiés dans la section 4.2 sont, cette fois, testés sur les instances issues de notre générateur et correspondant à chacun des scénarii proposés. Le taux de remplissage a été optimisé via la méthode directe. Les résultats se trouvent dans le tableau 6.2 et sont très semblables à ceux obtenus sur les instances issues de la littérature. La relaxation linéaire est toujours de bonne qualité ($MP\% \approx 0$) et amène rapidement à une solution entière ; sauf pour l'instance *dc201* et le critère de distance, où 539 nœuds sont créés. L'optimisation suivant le taux de remplissage fournit des solutions aux valeurs proches de celles fournies par l'optimisation de la distance sur un 1-graphe. Cependant, les différences constatées sur le nombre de véhicules utilisés pour ces deux optimisations suggèrent certaines différences structurelles qui pénalisent plus les trajets regroupant des requêtes en vue d'un gain en distance parcourue et qui génèrent un faible taux de remplissage. Nous remarquons qu'au plus la demande est diffuse (dans le temps et le territoire), moins la réduction des fenêtres de temps (de un tiers à un quart du chemin le plus rapide) a d'impact sur le nombre de véhicules utilisés et la distance parcourue. Ces résultats montrent que même sur un réseau réel, la modélisation par un *p*-graphe améliore significativement les résultats en terme de distance totale parcourue.

6.1. Plate-forme de travail couplée à un SIG

nom critère	veh.	dist.	tps. perdu	tx remp.	durée	roulée	MP%	nœ.	sec.
dc199 dist. alt.	11	1463,6	122,8	0,6599	4623,2	1230,9	0,0	1	1126
dist.	11	1592,1	43,5	0,6657	4225,9	1119,5	0,0	5	42
tx remp.	13	1597,8	53,0	0,6480	4124,3	1119,7	0,1	1	12
tps. perdu	11	2178,5	0,0	0,9013	3703,3	1466,1	0,0	2	1
dc200 dist. alt.	16	1273,3	77,3	0,9141	3617,8	1169,5	0,0	1	174
dist.	10	1425,0	30,9	0,9098	3190,2	1064,8	0,0	11	28
tx remp.	17	1425,9	27,6	0,9083	2906,8	1064,4	0,0	1	7
tps. perdu	10	1650,4	0,0	1,0801	3357,4	1230,3	0,0	10	2
dc201 dist. alt.	14	1173,2	187,5	0,6394	1199,9	929,2	1,5	87	1724
dist.	13	1253,2	131,2	0,6171	989,7	835,5	1,3	539	216
tx remp.	14	1264,8	152,9	0,5947	1098,4	844,2	0,5	15	15
tps. perdu	13	1956,3	5,3	1,0064	1454,0	1255,9	0,0	3	4
dc202 dist. alt.	11	1155,0	148,1	0,5239	3810,4	865,1	0,0	1	3578
dist.	13	1350,5	70,0	0,5212	3464,2	779,0	0,0	19	30
tx remp.	12	1354,8	75,8	0,5133	3431,5	781,7	0,0	1	14
tps. perdu	13	1928,8	0,0	0,7662	3929,9	1095,8	0,0	4	1
dc374 dist. alt.	11	1471,4	101,2	0,6727	4616,6	1217,1	0,0	1	674
dist.	12	1596,2	25,2	0,6820	4129,2	1120,6	0,0	1	18
tx remp.	11	1599,8	20,4	0,6775	4231,2	1117,7	0,0	1	15
tps. perdu	11	2238,5	0,0	0,9222	3960,2	1500,2	0,0	15	1
dc375 dist. alt.	16	1323,0	43,0	0,9681	3922,7	1188,7	0,0	3	197
dist.	16	1467,1	12,2	0,9615	3644,6	1096,7	0,0	1	6
tx remp.	18	1468,0	10,6	0,9604	3970,9	1096,2	0,0	1	8
tps. perdu	13	1698,2	0,0	1,1149	3332,4	1269,9	0,0	13	1
dc376 dist. alt.	14	1361,3	118,4	0,7667	1362,1	1066,8	0,0	1	140
dist.	16	1457,5	65,9	0,7363	1212,9	960,0	0,0	1	5
tx remp.	15	1464,5	73,2	0,7180	1316,7	965,1	0,0	1	5
tps. perdu	16	2122,8	0,0	1,0915	1671,7	1359,0	0,0	2	2
dc377 dist. alt.	14	1278,0	101,6	0,5916	4192,6	952,4	0,0	1	2828
dist.	16	1494,9	21,7	0,5897	4241,4	855,2	0,0	1	14
tx remp.	15	1497,8	38,9	0,5839	4198,0	856,8	0,0	1	13
tps. perdu	14	2086,5	0,0	0,8286	3755,7	1185,1	0,0	19	1

TAB. 6.2: Résultats des 3 critères de QoS pour 8 instances réalistes

6.2 Étude de la sinuosité

La forme est un sujet d'étude privilégié des géographes. Les analyses concernent des objets très variés comme des zones urbaines, des parcelles agricoles, des cours d'eau ou des flux de population. Le sujet d'étude le plus courant pour la sinuosité est l'écoulement de l'eau, notamment pour les géologues qui étudient les dépôts de sédiments (Schumm, 1963). Plazanet (1996) et Barillot (1999) s'intéressent à la sinuosité des réseaux routiers mais dans un but de représentation et de modélisation des courbures. Dans sa thèse, Plazanet (1996) cherche des fonctions de généralisation d'objets linéaires avec comme exemple type les routes, adéquates avec l'échelle d'affichage souhaitée. Il en résulte une étude détaillée des différents types de virages qui sont directement liés à la notion de sinuosité. Cependant notre analyse et les indices de sinuosité que nous avons choisi se distinguent de cette étude sur deux points : la considération de tournées et non d'objets inertes comme les routes et la perspective d'optimiser via ces indices de sinuosité.

L'optimisation des tournées de véhicule est rarement liée à la forme réelles des tournées, notamment car elles sont généralement calculées à partir de données agrégées « déspatialisées ». Pourtant trois éléments nous font penser que certaines informations liées à la forme des tournées méritent d'être prises en compte. Le premier élément relève de la qualité de service directe offerte au passager qui supporte mal de passer plusieurs fois au même endroit, de tourner en rond, de se rapprocher puis de s'éloigner de son lieu de destination ou d'en passer à proximité sans être déposé, de voir passer un véhicule du service à son arrêt sans le ramasser, ou bien dans le domaine du confort physique, de subir de nombreux virages serrés. Le deuxième élément nous incitant à nous intéresser à la forme des tournées, concerne l'allure générale d'un ensemble de tournées composant une solution. Il semble que la forme générale puisse induire un critère d'évaluation global d'une solution. Par exemple dans le cas de problèmes de transports convergents vers un centre ville, on peut estimer que les bonnes tournées sont radiales et non spiroïdales. À ce sujet, Lu et Dessouky (2006) proposent de minimiser le nombre de croisements inter-tournées. Notons que ce type d'analyse peut permettre de limiter l'espace des solutions envisagées lors d'une résolution heuristique. La répartition équitable sur le réseau des véhicules, qui est un enjeu important pour les systèmes de transport en temps réel ou la répartition doit être en adéquation avec des prévisions de demandes, peut aussi être intégrée à la définition de la forme des tournées. De telles considérations posent le problème de la représentation de données spatio-temporelles dans un SIG, dont on trouve un état de l'art dans Abraham et Roddick (1999). Ce problème a deux variantes. Dans la version la plus classique, l'observateur est en retrait des objets mouvants (Güting et al., 2000). L'autre situation, traitée dans Noyon et al. (2007), est celle d'un observateur qui est lui aussi un objet mouvant, dans le cas des TAD cela pourrait être un chauffeur (Noyon et al., 2007).

Ce problème se rencontre dans de nombreuses applications, comme pour la représentation des trajectoires dans l'espace

Le troisième motif d'incitation à cet étude est de fournir les moyens aux géographes

de réfléchir aux interactions du système de transport avec son environnement à travers la forme des tournées générées en fonction de la demande.

L'aspect de la forme des tournées auquel nous nous intéressons est leur sinuosité des tournées. Nous poursuivons deux objectifs principaux liés à différentes échelles d'observation des tournées. Tout d'abord, nous souhaitons réduire l'inconfort physique en évitant les tronçons de route contenant un grand nombre de virages sur peu de distance. Nous souhaitons ensuite prendre en compte l'inconfort mental provoqué par un itinéraire n'ayant pas pour direction générale la destination du passager. La sinuosité ayant une partie de sa définition assez vague et subjective, il est nécessaire d'en donner une ou des définitions formelles et d'en évaluer la pertinence.

6.2.1 Définition de quatre critères d'évaluation

Comme nous l'avons déjà dit, les géographes se sont d'avantage intéressés à la forme des objets que les chercheurs développant les algorithmes fournissant les tournées. Parmi la littérature traitant de la mesure de la sinuosité, nous avons principalement trouvé des travaux ayant pour sujet les cours d'eau. Les deux mesures utilisées pour des cas généraux sont : l'efficacité ou rectilinéarité, que nous présentons dans la section 6.2.1.1 et la dimension fractale (Nikora, 1991) dont l'interprétation en terme de qualité de service, ne nous a pas paru évidente dans le cas des tournées de véhicules. Quelques autres méthodes *ad hoc* basées sur des mesures angulaires sont proposées pour traiter des cas particuliers. Ces mesures de sinuosité se font généralement à partir de projections dans des espaces à deux dimensions. Un biais, peut alors apparaître avec l'utilisation de données (longueurs) provenant d'un espace à trois dimensions. Dans cette section, nous définissons quatre mesures différentes de la sinuosité d'une courbe.

Dans les données fournies par l'IGN, les routes ont pour élément atomique des segments de route. Les formulations proposées tiennent compte de cette approximation des routes par des courbes linéaires par morceaux. Ainsi, une courbe \mathcal{C}_{ij} est une suite d'arcs (a_0, a_1, \dots, a_n) dont la longueur respective est notée D_{a_i} et dont les coordonnées sont notées $((x_i, y_i), (x_{i+1}, y_{i+1}))$, et qui forment une chaîne de i à j . La distance à vol d'oiseau entre deux points i et j est notée d_{ij} .

6.2.1.1 Efficacité

L'efficacité est le critère le plus fréquemment utilisé pour mesurer la sinuosité d'une courbe. Elle se définit comme le rapport de la longueur de la courbe étudiée avec une longueur idéale (Gutiérrez et al., 1998). Dans le cas d'un trajet en voiture, on divise la longueur du plus court chemin à vol d'oiseau (ou éventuellement par le réseau routier) séparant deux points par la distance réellement parcourue entre ces deux points, comme dans l'équation (6.1). La figure 6.4 illustre cette mesure pour la course relative à la requête r . Il est clair que les valeurs de l'efficacité sont comprises entre 0 et 1.

$$eff(r^+, r^-) = \frac{d_{r^+r^-}}{\sum_{a \in \mathcal{C}_{r^+r^-}} D_a} \quad (6.1)$$

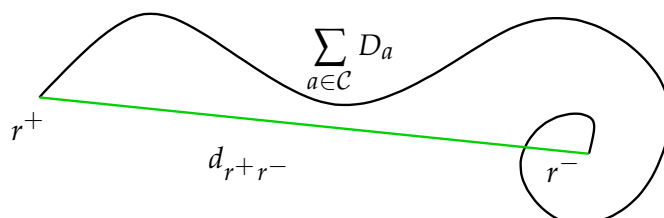


FIG. 6.4: Efficacité d'une course

Mesurer l'efficacité globalement sur une tournée comme nous l'avons définie n'a pas de sens ; surtout si le véhicule part et arrive au même dépôt, une division par zéro apparaît alors dans la formulation. Néanmoins, l'efficacité peut être mesurée pareillement sur chaque portion séparant deux arrêts consécutifs.

Cet indicateur est assez grossier, il peut donner la même valeur à une sinusoïde qu'à un coude. En terme d'optimisation (maximisation), la maximisation de l'efficacité s'apparente à la minimisation de la somme des distances parcourues pour chaque course.

Il est aussi remarquable que l'efficacité tient compte de la sinuosité verticale au même titre que de la sinuosité horizontale alors qu'en voiture ces deux types de mouvements ne sont pas ressentis identiquement par les passagers.

6.2.1.2 Tours de Volant

Ce critère est le plus naturel pour mesurer la sinuosité d'un trajet en voiture. Il correspond à l'intégration tout au long de la courbe, de l'angle $\alpha(t)$ entre la tangente à la direction en t et la direction en $t + \Delta t$. Dans le cas de courbes linéaires par morceaux sa définition mathématique est donnée par les équations (6.2) et (6.3).

$$\alpha(a_i a_{i+1}) = \begin{cases} \pi - \widehat{a_i a_{i+1}} & \text{si } \widehat{a_i a_{i+1}} \in [0, \pi] \\ \widehat{a_i a_{i+1}} - \pi & \text{si } \widehat{a_i a_{i+1}} \in]\pi, 2\pi] \end{cases} \quad (6.2)$$

$$tourv(r^+, r^-) = \sum_{a_i \in \mathcal{C}_{r^+r^-} \setminus \{a_n\}} \alpha(a_i a_{i+1}) \quad (6.3)$$

Ce critère mesure uniquement la sinuosité horizontale d'une course. La sinuosité en tours de volant d'une tournée se définit à partir de la même formule. Le nombre de tours de volant ne tient pas compte directement de la distance parcourue, donc de la densité des virages.

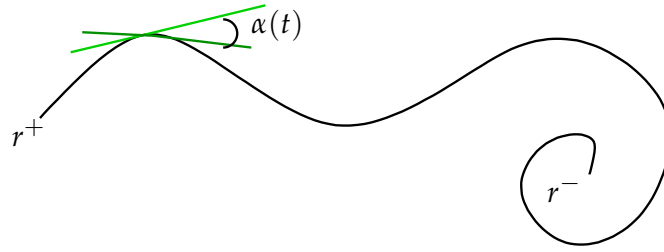


FIG. 6.5: Quantité de tours de volant d'une course

6.2.1.3 Distance à la destination

Afin d'intégrer à la notion de sinuosité, la perception de la pertinence de l'itinéraire qu'en ont les passagers, nous proposons d'évaluer la distance à la destination (de coordonnées (x_{n+1}, y_{n+1})) en chaque point de la course par une intégrale ayant pour bornes le ramassage et la livraison. Dans le cas de courbes linéaires par morceaux, on obtient l'équation (6.6). Où les $x_{i(t)}$ et les $y_{i(t)}$ définis par les équations (6.4) et (6.5) sont les coordonnées du point $(x_{i(t)}, y_{i(t)})$ variant le long du segment $((x_i, y_i), (x_{i+1}, y_{i+1}))$. La formulation présentée ici permet l'application directe des méthodes de quadrature de Gauss pour le calcul numérique d'intégrales.

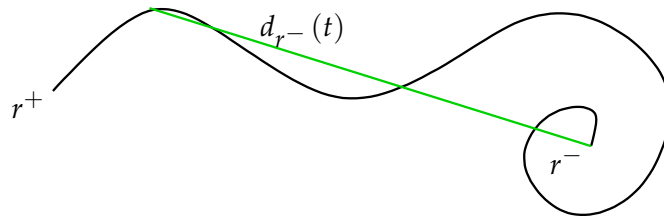


FIG. 6.6: Distance à la destination au long d'une course

$$x_{i(t)} = x_i + (t + 1) \frac{x_{i+1} - x_i}{2} \quad (6.4)$$

$$y_{i(t)} = y_i + (t + 1) \frac{y_{i+1} - y_i}{2} \quad (6.5)$$

$$distdest(r^+, r^-) = \sum_{a_i \in \mathcal{C}_{r^+ r^-}} \int_{-1}^1 \sqrt{(x_{n+1} - x_{i(t)})^2 + (y_{n+1} - y_{i(t)})^2} dt \quad (6.6)$$

La distance à la destination mise sous cette forme, pénalise plus fortement les détours éloignés de la destination que ceux effectués à proximité ; ce qui est l'effet inverse de l'effet recherché. Par contre, la dérivée de cette fonction, c'est-à-dire la vitesse à laquelle évolue la distance à la destination n'a pas cet inconvénient. Le problème de la vitesse est qu'elle peut changer de signe ce qui entraîne des effets de compensation lors du calcul d'une intégrale.

6.2.1.4 Distance angulaire

Dans le souci de pénaliser plus fortement les écarts à la direction directe vers la destination en fonction de leur proximité de la destination, nous proposons une quatrième mesure de la sinuosité que nous appelons distance angulaire. Il s'agit de mesurer en tout point de la courbe, l'angle $\theta(t)$ entre la tangente à la trajectoire et la ligne droite vers la destination, comme l'illustre la figure 6.7. Afin de régler des défauts liés à la symétrie des angles orientés positivement avec ceux orientés négativement, nous considérons l'angle $\theta(t)/2$ et nous intégrons la fonction $1 - \cos(\theta(t)/2) \in [0, 1]$, avec $\theta(t) \in]-\pi, \pi]$. Ce qui donne la formule (6.8), une fois exprimée³ suivant les coordonnées de la courbe $\mathcal{C}_{r^+r^-}$ (voir équation (6.7)) avec θ_i la fonction exprimant $\theta(t)$ sur l'arc a_i .

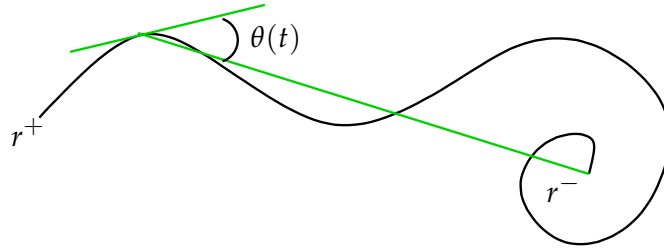


FIG. 6.7: Distance à la direction au long d'une course

$$\cos(\theta_i(t)) = \frac{(x_{n+1} - x_{i(t)})(x_{i+1(t)} - x_{i(t)}) + (y_{n+1} - y_{i(t)})(y_{i+1(t)} - y_{i(t)})}{\sqrt{(x_{i+1(t)} - x_{i(t)})^2 + (y_{i+1(t)} - y_{i(t)})^2} \sqrt{(x_{n+1} - x_{i(t)})^2 + (y_{n+1} - y_{i(t)})^2}} dt \quad (6.7)$$

$$angledest(r^+, r^-) = \sum_{a_i \in \mathcal{C}_{r^+r^-}} \int_{-1}^1 1 - \cos(\theta_i(t)/2) \quad (6.8)$$

6.2.2 Observations

Nous présentons les valeurs de ces indices sur trois exemples, les deux premiers, représentés sur la figure 6.8 sont des cas simplistes mais représentatifs des problèmes rencontrés sur la perception des tournées par les passagers. Le premier exemple compare les courbes b1 et b2 qui ont la même longueur, mais un nombre différent de points d'inflexions. Les courbes c1 et c2 sont symétriques et forment un détour de même forme soit à l'origine, soit à la destination. Le troisième exemple considère un cas réel dans le Doubs Central. Les courbes a1 et a2 correspondent respectivement au plus court chemin sur le réseau routier entre le ramassage v_{19+} (personnages grossis) et la livraison v_{19-} (gros point) qui est en pointillés sur la figure 6.10, et à la portion du trajet (en trait plein sur la figure 6.10) suivie par la tournée de la figure 6.9. Ce trajet passe par v_{18-}

³Nous rappelons la formule trigonométrique $\cos(\theta(t)) = \sqrt{(1 + \cos(\theta(t)))/2}$.

(petit point), ce qui oblige un détour qui induit un gain de temps et permet de satisfaire les contraintes horaires des requêtes 18 et 19.

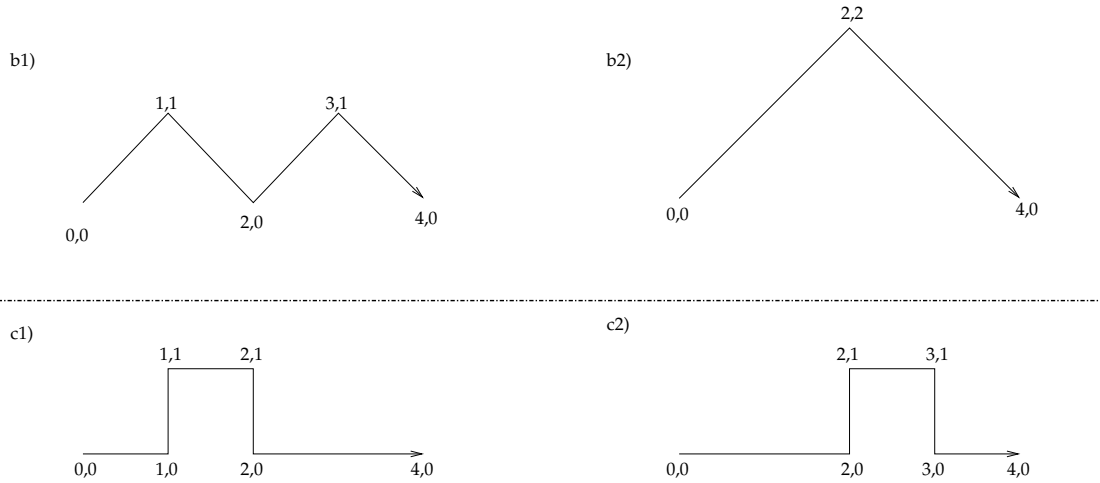


FIG. 6.8: Exemples de courbes

courbe	efficacité	tours de volant	dist. destination	dist. angulaire
b0	1,00	0,00	4,00	0,00
b1	0,71	4,71	16,88	0,63
b2	0,71	1,57	18,64	0,66
c0	1,00	0,00	4,00	0,00
c1	0,52	4,71	26,66	1,17
c2	0,52	4,71	23,08	1,18
a0	1,00	0,00	14,06	0,00
a1	0,83	43,21	16,14	17,55
a2	0,64	61,08	17,27	46,43

TAB. 6.3: Valeurs des indices de sinuosité

Hormis pour l'efficacité, les valeurs minimales sont synonymes de qualité (faible sinuosité). Dans le tableau de résultats 6.3, nous rajoutons les courbes a0, b0 et c0 qui correspondent à la ligne droite entre origine et destination. Nous remarquons que l'efficacité ne discerne pas les courbes b1, c1 des courbes b2 et c2. Le nombre de tours de volants lui ne distingue pas la localisation des détours par rapport à la destination, il est donc identique pour les courbes c1 et c2. La distance à la destination discrimine toutes les courbes, mais distingue aussi fortement les lignes droites suivant leur longueur, par exemple c0 avec a0 et b0. Enfin, la distance angulaire discrimine les courbes qui ont des changements de direction. De plus dans le cas des courbes c1 et c2, c'est le seul indice à fournir une meilleure valeur à la courbe c2 qui effectue ses principaux détours près de l'origine plutôt que près de la destination. Cette remarque s'applique aussi encore plus nettement sur le cas réel entre les courbes a1 et a2.

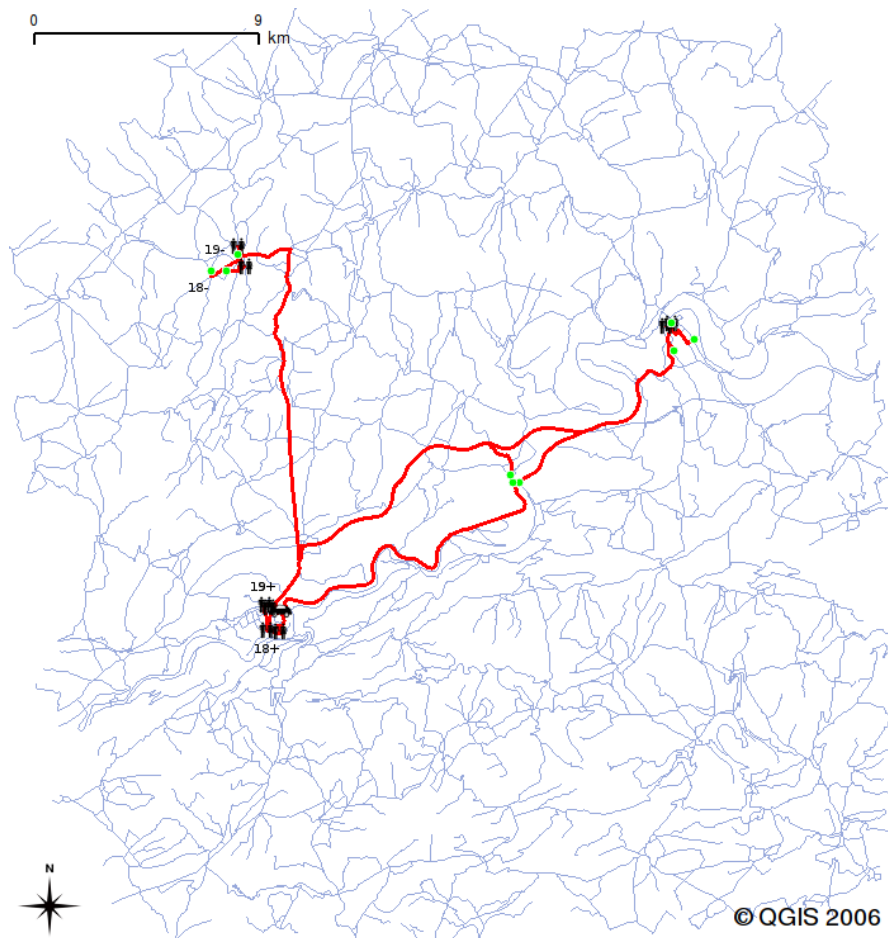


FIG. 6.9: Aperçu sur le réseau routier du Doubs Central d'une tournée extraite d'une solution

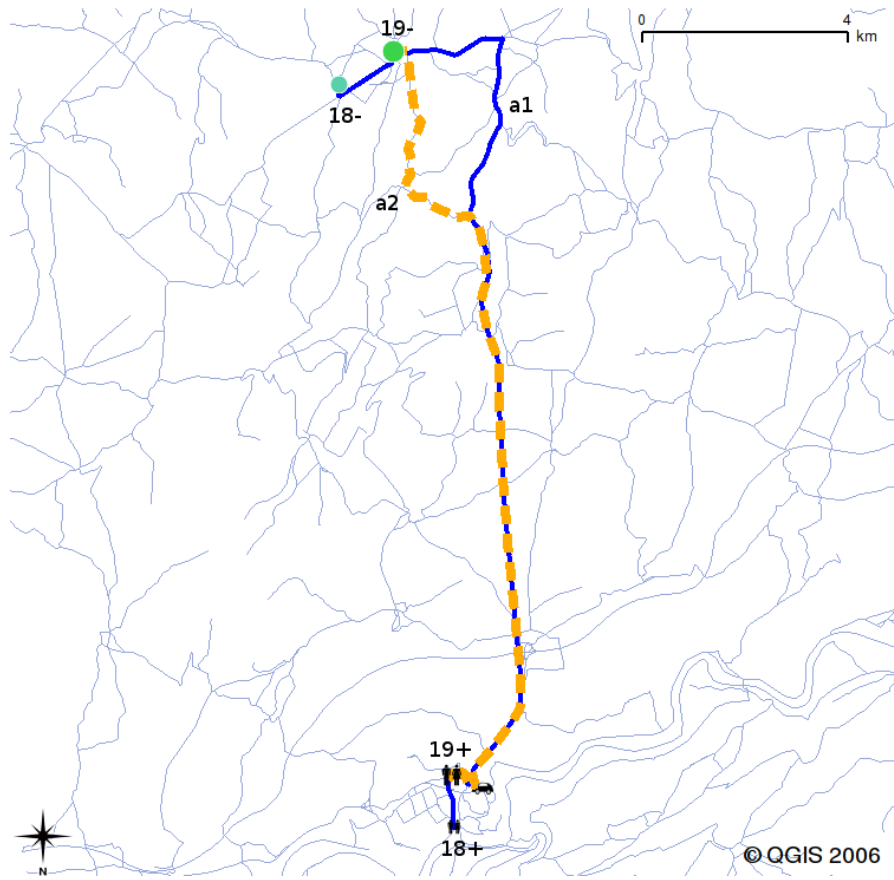


FIG. 6.10: Portion de route relatives aux requêtes 18 et 19

Conclusion

Déjà en pleine expansion lorsque nous avons débuté cette thèse, les systèmes de transport à la demande (TAD) ont poursuivi leur développement durant ces dernières années. Et le nombre élevé de projets en cours augure d'un avenir florissant pour les TAD. Cependant, leurs modes de fonctionnement ne sont pas encore arrivés à maturité. La minimisation des coûts d'exploitation et la maximisation de la qualité du service rendu, notamment en terme de souplesse, présentent encore de nombreux défis aux chercheurs et aux opérateurs. À cela vient s'ajouter le flou encore très présent autour de la place que les TAD prennent ou doivent prendre dans l'offre de transport faite aux citoyens.

Au sein d'une mission d'un projet de recherche-action mêlant collectivités locales, compagnies de taxis et des chercheurs principalement géographes et informaticiens, cette thèse a pour objectif de coupler une analyse qualitative géographique à une optimisation mathématique des tournées de véhicules. Un autre impératif de ce travail est de livrer un module de calcul des tournées pour un système de transport à la demande à réservation à *J-1*, appelé TADOU et implanté dans le Pays du Doubs Central. Ce TAD a servi de sujet expérimental au projet et après quelques ajustements en début de fonctionnement, donne à présent entière satisfaction aux collectivités locales l'exploitant. En pratique, le noyau d'optimisation de TADOU est basé sur une approche heuristique basée sur une procédure d'insertion tandis qu'une méthode exacte est utilisée pour l'analyse des critères. Les deux algorithmes sont couplés avec un Système d'Information Géographique (SIG) pour permettre l'analyse géographique des tournées. L'ensemble des produits permettant l'intégration du calcul des tournées de véhicules au SIG et le SIG lui-même sont des logiciels libres.

Une première réflexion est menée sur la notion de qualité de service que nous élargissons à tout critère pouvant nécessiter une optimisation lors de l'élaboration des tournées ; ceci étant fonction des différents acteurs des TAD (AOT, transporteur et usagers). Nous définissons ainsi de nombreux critères de qualité de service originaux. Une fois listés et classés par rapport à leur niveau d'évaluation (sur un tronçon, une course, une tournée, une ou plusieurs solutions), il apparaît qu'un grand nombre de ces critères peuvent s'évaluer à un niveau assez bas, par exemple les tronçons, et se cumuler pour être évalués aux niveaux supérieurs. Ceci s'explique d'une part par le fait, qu'en général, le désagrément subi par un passager est uniquement lié à la tournée où il se trouve dans le véhicule, sans tenir compte des autres tournées.

Dans un contexte où on minimise l'inconvénient global subi par les passagers, la méthode de résolution par décomposition de génération de colonnes semble appropriée puisqu'elle permet de considérer un sous-problème (ou problème esclave) évaluant de façon indépendante chaque tournée parmi lesquelles un problème principal (ou problème maître) permet leur sélection dans une solution optimale par rapport à un inconvénient global cumulant (par une somme) les inconvénients liés à chaque tournée. D'ailleurs, c'est sur cette approche que sont basées beaucoup de méthodes, parmi les plus efficaces, résolvant de manière exacte ou heuristique des problèmes de calcul de tournées à plusieurs véhicules.

Nous avons donc développé un algorithme de génération de colonnes dans lequel nous avons intégré séparément la minimisation de la distance parcourue, la minimisation du temps perdu et la maximisation du taux de remplissage des véhicules. En se restreignant à des requêtes définies par une date d'arrivée souhaitée et non dépassable pour le temps perdu, ces deux critères de qualité de service ont la propriété de pouvoir se calculer en cumulant leurs valeurs lors de la construction des tournées de proche en proche. Un algorithme de programmation dynamique est alors tout désigné pour résoudre ce sous-problème mis sous forme d'un problème de plus court chemin avec contraintes de ressources, tandis que le problème principal, prenant la forme d'un problème d'affectation généralisé qui sélectionne un ensemble de tournées satisfaisant toutes les requêtes, est résolu par l'algorithme du simplexe. Dans le cas de la distance et du temps perdu, le problème principal est directement résolu en faisant appel à un solveur du commerce (Cplex). Le cas du taux de remplissage présente, dans sa forme originelle, une fonction objectif fractionnaire linéaire ne s'exprimant pas comme la somme des coûts des tournées. Nous proposons donc une modification du problème principal, soit dans la fonction objectif, soit par l'ajout de contraintes. Les impacts de ces modifications sont alors ressentis au niveau de la résolution du sous-problème, les coûts ne respectant plus certaines propriétés. Afin d'être raisonnablement efficace – quelques secondes de calcul pour des instances à une cinquantaine de requêtes –, la mise en œuvre de l'algorithme de programmation dynamique fait l'objet de plusieurs développements spécifiques et paramétrages, dont nous présentons ceux s'étant révélés les plus pertinents et permettant de conserver un fort potentiel de généralité. Ces développements ont été testés sur des instances de PDPTW de la littérature, des instances de DARP qui en sont dérivées ainsi que des instances réalistes, c'est-à-dire issues de demandes réelles sur un véritable réseau routier. Bien que n'étant pas la plus rapide, notre méthode parvient à résoudre exactement des instances d'une centaine de requêtes en quelques minutes. Mais plus important encore, la méthode s'adapte avec succès aux trois critères choisis ; ce qui correspond au caractère générique souhaité pour la méthode.

Le fait de plonger les tournées dans le réseau réel nous a amené à développer une optimisation des tournées dans un p -graphe qui prend en compte les chemins alternatifs offerts par le réseau routier. Malgré la forte augmentation de la taille des données, nous avons montré que la résolution dans un p -graphe est possible par programmation dynamique sur des instances réelles d'une zone rurale – le Pays du Doubs Central – dans des délais de calcul raisonnables. L'utilisation innovante de p -graphes dans le domaine du calcul de tournées de véhicules est prometteuse quant à l'amélioration des

solutions obtenues et quant à l'intégration de critères de qualité de service originaux. L'intégration d'éléments encore plus précis liés au réseau routier et à l'environnement est une voie intéressante en terme d'optimisation et d'observation géographique. Mais considérer des caractéristiques complexes des itinéraires comme celles liées à leur représentation spatiale génère, outre une très forte augmentation des données, des problèmes d'évaluation intra et inter tournées ; ce qui risque d'entraîner une modélisation mathématique complexe. Nous avons traité le cas de l'évaluation de la forme des tournées par des aspects de sinuosité. Quatre indices sont proposés sur cette notion. Cette étude met à profit le couplage du calcul de tournées avec les outils de SIG.

La décomposition du problème de DARP proposée se révèle pertinente pour la plupart des critères de qualité de service envisagés, surtout ceux liés à la satisfaction des passagers. L'efficacité de la méthode peut être accrue par l'utilisation de méthodes heuristiques au niveau du problème esclave. Cependant, la résolution exacte par programmation dynamique peut être mise en œuvre de façon à obtenir elle aussi des solutions assez rapidement. L'ajout de critères (donc de ressources) et de contraintes supplémentaires pénalise l'utilisation de règles de dominance fortes, comme nous l'avons effectuée, qui deviennent complexes et rarement vérifiées. Il paraît préférable de se concentrer sur les règles d'extension des tournées en construction qui sont soumises à de nombreuses contraintes dans le cas de contraintes dures et sur l'ordre de traitement de ces tournées partielles dans le cas de contraintes douces. Définir un ordre de traitement des requêtes revient à appliquer une règle de dominance incomplète et donc à repousser le traitement des tournées partielles ainsi incomplètement dominées. Dans le cas d'un nombre important de contraintes, la résolution du problème esclave semble pertinente par une autre approche exacte qu'est la programmation par contraintes, déjà utilisée avec succès pour le VRPTW (Rousseau et al., 2004; Rousseau, 2004).

Même si lorsque l'évaluation de la qualité des tournées est calculée indépendamment, celle d'une solution est très souvent (dans le cas des TAD) faite en sommant le niveau de qualité de service atteint par chaque tournée, il est envisageable dans le cadre des transports en commun de chercher à maximiser la qualité de service offerte aux passagers les moins bien servis. L'impact d'une telle modification du modèle d'optimisation mérite d'être observé en détails en terme d'efficacité de la méthode et de caractéristiques des solutions obtenues.

Notre étude s'est limitée à l'optimisation indépendante de chaque critère traité. Une piste d'amélioration globale de la qualité de service est de combiner plusieurs critères d'optimisation avec l'objectif de former des sous-ensembles de critères minimaux permettant de couvrir une qualité de service définie sur plusieurs critères, éventuellement non directement intégrés à l'optimisation. Les critères de qualité de service dépendant de la forme des tournées dans le réseau de transport sont une piste sérieuse pour répondre à ce problème, puisqu'ils permettent de traduire une vision globale de l'ensemble des solutions.

Les TAD sont des systèmes totalement plongés dans un territoire, le déroulement du service en devient très volatile car il met en scène des utilisateurs et un réseau routier non maîtrisés auxquels on peut ajouter des perturbations liées aux véhicules et aux

chauffeurs. Ceci pose des problèmes de robustesse du système ; ce qui se traduit pour le calcul des tournées dans le cas statique à des problèmes de robustesse des solutions. Des perturbations au niveau des données comme l'allongement arbitraire des durées de transport, sont généralement opérées pour prévenir d'éventuels impondérables. Ces problèmes de robustesse se retrouvent pour les TAD en temps réel ou dynamiques, c'est-à-dire avec un délai de réservation permettant l'apparition ou la disparition de requêtes lorsque les véhicules disponibles sont déjà en mouvement. Ce type de système tend à devenir un modèle avec des délais de réservation – avec un taux d'acceptation des requêtes élevé – les plus brefs possibles.

La question de la pertinence pour le cas dynamique de la méthode proposée dans cette thèse pour le cas statique mérite d'être posée. Deux types de fonctionnement sont séparables. Le premier type correspond à un système où aucune (ou presque) requête n'est connue avant le démarrage du service. Ici, on calcule des tournées pour servir des requêtes et finir de servir celles qui sont en cours, les véhicules étant positionnés suivant la solution précédente. Il est raisonnable d'espérer que ce problème de taille restreinte soit résolu rapidement par notre méthode. Plusieurs arguments nous amènent à cette conjecture. Tout d'abord, à l'arrivée d'une nouvelle requête, nous disposons déjà d'un ensemble de tournées valides pour tout le reste des services à effectuer. Ces tournées peuvent être réduites à un intervalle de temps ou sous-ensemble de véhicules (en fonction de leur position géographique) pour lesquels l'insertion de la nouvelle requête est susceptible d'avoir un impact réel ; il n'est par exemple plus nécessaire de considérer les services déjà effectués. De plus, les contraintes de fenêtre de temps sont souvent fortement réduites pour les requêtes dont la réservation a déjà été validée et pour les requêtes en cours, le véhicule est déjà fixé et l'horaire de livraison est souvent soumis à une contrainte supplémentaire de durée de transport maximale. Comme nous avons pu le constater expérimentalement la diminution du nombre de services et l'augmentation de ce type de contraintes a tendance à accélérer la résolution du problème via notre approche.

Même si notre approche peut répondre rapidement à une demande dynamique, nous n'apportons aucune garantie sur la probabilité d'acceptation des requêtes. Pour cela, il faut imaginer une évaluation pertinente de la faculté pour une solution courante d'accepter de nouvelles requêtes. Ce critère dépend fortement de l'environnement (connaissance de prévisions stochastiques sur l'apparition des demandes, étalement de la zone à couvrir par le TAD. . .) mais une fois défini il peut être intégré au modèle d'optimisation sous forme d'objectifs ou de contraintes. Ce critère concerne naturellement l'ensemble des tournées ; cela risque d'être aussi le cas pour son évaluation. Ce type de critère, comme nous l'avons montré pour le taux de remplissage, peut dans certains cas être traité efficacement par notre approche. Répondre à ce problème permet de répondre à celui proposé par le deuxième type de fonctionnement évoqué où une partie des requêtes est connue par avance.

La méthode par génération de colonnes, même dans une version menant à une solution strictement optimale, nous paraît mériter d'être considérée dans le cadre de TAD dynamiques car nous pensons possible d'obtenir un coût associé en temps de calcul, négligeable.

Liste des illustrations

1.1	Représentation du Transport à la Demande - Les demandes	11
1.2	Représentation du Transport à la Demande - Les Tournées	11
1.3	Carte des Systèmes de Transport Intelligents aux États-Unis par type de technologie.	13
1.4	Niveaux de compétence des AOT en France	17
2.1	Méthode de génération de colonnes.	33
2.2	Un SPPRC à une ressource	37
2.3	Trace de l'algorithme de programmation dynamique	41
3.1	Tournée ω du véhicule $k(\omega) = 2$	47
3.2	Représentation de la pénalité associée à l'écart par rapport à des horaires idéaux de début de service.	49
3.3	Représentation de la pénalité associée à la durée d'un trajet par des fonctions du temps associées à chaque arrêt	50
3.4	Temps d'attente absurdes	50
3.5	Ordonnancement des services pour un véhicule en un même lieu entre deux trajets	52
3.6	Décomposition de la durée d'une course	53
3.7	Modification de la course à partir du réseau	57
3.8	Deux types de boucles	58
3.9	Tournées avec et sans croisements	59
3.10	Deux tournées en conflit	60
3.11	Fonction d'inconvénient liée aux dates de service (Sexton et al.).	63
3.12	Fonction d'inconvénient liée aux dates de service (Dumas et al.).	64
3.13	Utilisation de chemins alternatifs	66
3.14	Algorithme de programmation dynamique pour le Problème de Sélection des Arcs dans une Séquence Fixée	69
4.1	Méthode de Séparation et Évaluation Progressive par Génération de Colonnes	77
4.2	Inégalité triangulaire conservée dans EP au niveau des livraisons	82
4.3	Coût du détour par k_1 et k_2 de i vers j	84
4.4	Les six configurations pour servir deux requêtes	85
4.5	Recherche de chemins élémentaires par un LDS paramétré à 1 bon voisin	87

4.6	Résolution heuristique du problème esclave	89
4.7	Résolution du problème esclave par programmation dynamique	90
4.8	Les 3 classes d’instances de Li et Lim	92
4.9	Adaptation de la méthode à la distance totale parcourue	103
4.10	L’algorithme de Dinkelbach	108
4.11	L’algorithme de Dinkelbach avec la génération de colonnes – pour minimiser $-N/D$	109
4.12	Inégalité triangulaire pour la méthode itérative	110
4.13	Méthode directe intégrée au SEPGCOL – pour minimiser $-N/D$	112
4.14	Inégalité triangulaire pour la méthode directe	113
5.1	Densités de populations et migrations alternantes dans le Pays du Doubs Central (Josselin, 2005)	128
5.2	Accueil de TADOU	133
5.3	Enregistrement d’une requête avec paramétrage de son niveau de qualité de service	133
5.4	Résolution en trois phases	135
5.5	Insertion d’une requête dans une séquence – construction du p -graphe	136
5.6	Quatre solutions typées pour des instances à 50 requêtes dans le Doubs Central	140
6.1	Plate-forme d’édition, de visualisation et de diffusion de tournées de véhicules	145
6.2	Regroupement sans détours dans un réseau routier	146
6.3	Réseau routier du Doubs Central	147
6.4	Efficacité d’une course	152
6.5	Quantité de tours de volant d’une course	153
6.6	Distance à la destination au long d’une course	153
6.7	Distance à la direction au long d’une course	154
6.8	Exemples de courbes	155
6.9	Aperçu sur le réseau routier du Doubs Central d’une tournée extraite d’une solution	156
6.10	Portion de route relatives aux requêtes 18 et 19	157

Liste des tableaux

1.1	Comparatif entre différents modes de transport	15
2.1	Récapitulatif des caractéristiques de quelques problèmes classiques	22
3.1	Classification des critères de qualité de service par niveau d'agrégation	48
4.1	Résultats sur les instances de Li et Lim à ≈ 50 requêtes	93
4.2	Caractéristiques des instances de Li et Lim originales et modifiées (en gras)	95
4.3	Caractéristiques des instances de Cordeau originales et modifiées (en gras)	96
4.4	Résultats pour les instances de Li et Lim modifiées	97
4.5	Résultats pour les instances 'b' de Cordeau modifiées	98
4.6	Valeurs maximales de f_i pour $k \in \mathcal{R}^-$	110
4.7	Valeurs maximales de f_i pour $k \in \mathcal{R}^+$	110
4.8	Caractéristiques des instances de Li et Lim modifiées par un p -graphe	114
4.9	Caractéristiques des instances de Cordeau modifiées par un p -graphe	115
4.10	Résultats pour les instances de Li et Lim modifiées – gr101-gr106	116
4.11	Résultats pour les instances de Li et Lim modifiées – gr107-gr112	117
4.12	Résultats pour les instances de Li et Lim modifiées – gc101-gc109	118
4.13	Résultats pour les instances de Li et Lim modifiées – grc101-grc108	119
4.14	Résultats des 3 critères de QoS pour instances de Cordeau modifiée –1	120
4.15	Résultats des 3 critères de QoS pour instances de Cordeau modifiée –2	121
4.16	Résultats des 3 critères de QoS pour instances de Cordeau modifiée –3	122
5.1	Résultats pour huit instances réalistes	141
5.2	Résultats de l'heuristique pour les instances 'gr' et 'mr'	141
5.3	Résultats de l'heuristique pour les instances 'gc' et 'mc'	142
5.4	Résultats de l'heuristique pour les instances 'grc' et 'mrc'	142
6.1	Caractéristiques de 8 instances réalistes	148
6.2	Résultats des 3 critères de QoS pour 8 instances réalistes	149
6.3	Valeurs des indices de sinuosité	155

Bibliographie

- (Abraham et Roddick, 1999) T. Abraham et J. Roddick, 1999. Survey of spatio-temporal databases. *Geoinformatica* 3(1), 61–99.
- (Ahuja et al., 1998) R. Ahuja, R. Dial, et J. Orlin, 1998. Fast infeasibility detection algorithms for dial-a-ride transit problems. Rapport technique WP 4005-98, Massachusetts Institute of Technology (MIT), Sloan School of Management. <http://ideas.repec.org/p/mit/sloanp/2699.html>.
- (Ahuja et al., 2003) R. Ahuja, D. Hochbaum, et J. Orlin, 2003. Solving the convex cost integer dual network flow problem. *Management Science* 49(7), 950–964.
- (Ambrosino et Nelson, 2004) G. Ambrosino et J. Nelson, 2004. *Demand Responsive Transport Services : Toward the Flexible Mobility Agency*. Italy : ENEA.
- (Ascher, 2002) F. Ascher, 2002. Le transport à la demande : individualisation des mobilités urbaines et personnalisation des services publics. *Annales des télécommunications* 57(3-4), 277–288.
- (Ascheuer et al., 2000) N. Ascheuer, M. Jünger, et G. Reinelt, 2000. A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints. *Computational Optimization and Applications* 17, 61–84.
- (Assad et Golden, 1995) A. Assad et B. Golden, 1995. Arc routing methods and applications. Dans M. Ball, T. Magnanti, C. Monna, et G. Nemhauser (Réds.), *Network Routing*, Volume 8 dans *Handbooks in Operations Research and Management Science*, 375–483. Amsterdam, North-Holland : Informs.
- (Bailly et Heurgon, 2001) J.-P. Bailly et E. Heurgon, 2001. *Nouveaux rythmes urbains : quels transports ?* (Édition de l’Aube ed.). Société et territoire, Série prospective du présent. La Tour d’Aigues. 222 p.
- (Balas et al., 1995) E. Balas, M. Fischetti, et W. Pulleyblank, 1995. The precedence-constrained asymmetric traveling salesman polytope. *Mathematical Programming* 68, 241–265.
- (Baldacci et al., 2004) R. Baldacci, V. Maniezzo, et A. Mingozzi, 2004. An exact method for the car pooling problem based on lagrangian column generation. *Operations Research* 52(3), 422–439.

- (Barillot, 1999) X. Barillot, 1999. Linear automatic generalization : some tools for road shapes measurement. *Bulletin du Comité français de cartographie* ISSN 0755-7647.
- (Barnhart et al., 1998) C. Barnhart, L. Ellis, G. Nemhauser, M. Savelsbergh, et P. Vance, 1998. Branch-and-price : Column generation for solving huge integer programs. *Operations Research* 46(3), 316–329.
- (Baugh et al., 1998) J. Baugh, G. Krishna, R. Kakivaya, et J. Stone, 1998. Intractability of the dial-a-ride problem and a multiobjective solution using simulated annealing. *Engineering Optimization* 30, 91–123.
- (Beasley et Christofides, 1989) J. Beasley et N. Christofides, 1989. An algorithm for the resource constrained shortest path problem. *Networks* 19, 379–394.
- (Ben Amor et Desrosiers, 2006) H. Ben Amor et J. Desrosiers, 2006. A proximal trust-region algorithm for column generation stabilization. *Computers & Operations Research* 33, 910–927.
- (Ben Amor et al., 2006) H. Ben Amor, J. Desrosiers, et J. Valério de Carvalho, 2006. Dual-optimal inequalities for stabilized column generation. *Operations Research* 54(3), 454–463.
- (Bent et Van Hentenryck, 2006) R. Bent et P. Van Hentenryck, 2006. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research* 33(4), 875–893.
- (Bodin et Sexton, 1986) L. Bodin et T. Sexton, 1986. The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in Management Science* 2, 73–86.
- (Borndörfer et al., 2001) R. Borndörfer, M. Grötschel, et A. Löbel, 2001. Scheduling duties by adaptive column generation. Rapport technique, Konrad-Zuze-Zentrum für Informationstechnik Berlin.
- (Castex, 2007) É. Castex, 2007. *Le transport à la demande en France : De l'état des lieux à l'anticipation. Modélisation des caractéristiques fonctionnelles des TAD pour développer les modes flexibles de demain*. Thèse de Doctorat, Université d'Avignon et des Pays de Vaucluse. Soutenue le 22 novembre 2007, 480 pages.
- (Castex et Josselin, 2007) E. Castex et D. Josselin, 2007. Temporalités éclatées : la réponse des transports à la demande aux nouvelles formes de mobilité. *Espace Population et Sociétés*. 15 p.
- (Chabrier, 2006) A. Chabrier, 2006. Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research* 66(4), 2972–2990.
- (Charikar et Raghavachari, 1998) M. Charikar et B. Raghavachari, 1998. The finite capacity dial-a-ride problem. Dans les actes de *39th Annual IEEE Symposium on Foundations of Computer Science*.
- (Charnes et Cooper, 1962) A. Charnes et W. Cooper, 1962. Programming with linear fractional functions. *Naval research logistics quart* 9, 181–186.

- (Chevrier et al., 2006a) R. Chevrier, P. Canalda, P. Chatonnay, et D. Josselin, 2006a. Comparison of three algorithms for solving the convergent demand responsive transportation problem. Dans les actes de *ITSC'2006, 9th Int. IEEE Conf. on Intelligent Transportation Systems*, Toronto, Canada, 1096–1101.
- (Chevrier et al., 2006b) R. Chevrier, P. Canalda, P. Chatonnay, et D. Josselin, 2006b. An oriented convergent mutation operator for solving a scalable convergent demand responsive transport problem. Dans les actes de *ICSSSM'06, IEEE Int. Conf. on Service Systems and Service Management*, Troyes, France, 959–964. 6 pages.
- (Chrétienne, 1999) P. Chrétienne, 1999. Minimizing the earliness and tardiness cost of a sequence of tasks on a single machine. Rapport technique 1999-007, LIP6.
- (Chrétienne et Sourd, 2000) P. Chrétienne et F. Sourd, 2000. Scheduling with convex cost functions. Rapport technique, LIP6.
- (Colorni et al., 1996) A. Colorni, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, et M. Trubian, 1996. Heuristics from nature for hard combinatorial optimization problems. *International Transactions in Operational Research* 3, 1–21.
- (Cordeau, 2006) J.-F. Cordeau, 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research* 54(3), 573–586.
- (Cordeau et Laporte, 2003a) J.-F. Cordeau et G. Laporte, 2003a. The dial-a-ride problem (DARP) : Variants, modeling issues and algorithms. *4OR* 1, 89–101.
- (Cordeau et Laporte, 2003b) J.-F. Cordeau et G. Laporte, 2003b. A tabu search heuristic algorithm for the static multi-vehicle dial-a-ride problem. *Transportation Research B* 37, 579–594.
- (Cordeau et Laporte, 2008) J.-F. Cordeau et G. Laporte, 2008. The dial-a-ride problem : models and algorithms. à paraître dans *4OR*.
- (Cordeau et al., 2004) J.-F. Cordeau, G. Laporte, J.-Y. Potvin, et M. Savelsbergh, 2004. Transportation on demand. Rapport technique 25, CRT, Montréal.
- (Coslovich et al., 2006) L. Coslovich, R. Pesenti, et W. Ukovich, 2006. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research* 175, 1605–1615.
- (Crainic et Laporte, 1998) T. Crainic et G. Laporte, 1998. *Fleet Management and Logistics*. Boston, USA : Kluwer.
- (Croes, 1958) G. Croes, 1958. A method for solving traveling-salesman problems. *Operational Research* 6, 791–812.
- (Daganzo, 1978) C. Daganzo, 1978. An approximate analytic model of many-to-many demand responsive transportation systems. *Transportation Research* 12, 325–333.
- (Daganzo, 1984) C. Daganzo, 1984. Checkpoint dial-a-ride systems. *Transportation Research* 18B, 315–327.

- (Dantzig et al., 1954) G. B. Dantzig, D. R. Fulkerson, et S. M. Johnson, 1954. Solution of a large scale traveling salesman problem. *Operational Research* 2, 393–410.
- (Desaulniers, 2007) G. Desaulniers, 2007. Managing large fixed costs in vehicle routing and crew scheduling problems solved by column generation. *Computers & Operations Research* 34(4), 1221–1239.
- (Desaulniers et al., 2002) G. Desaulniers, J. Desrosiers, A. Erdmann, M. Solomon, et F. Soumis, 2002. VRP with pickup and delivery. Dans P. Toth et D. Vigo (Rédts.), *The Vehicle Routing Problem*, 225–242. Philadelphia : SIAM Monographs on Discrete Mathematics and Applications.
- (Desaulniers et al., 2005) Desaulniers, G., J. Desrosiers, et M. Solomon (Rédts.), 2005. *Column Generation*. Springer.
- (Desrochers, 1988) M. Desrochers, 1988. An algorithm for the shortest path problem with resource constraints. Rapport technique, Les cahiers du GERAD, HEC, Montréal.
- (Desrochers et al., 1992) M. Desrochers, J. Desrosiers, et M. Solomon, 1992. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* 40(2), 342–354.
- (Desrosiers et al., 1995) J. Desrosiers, Y. Dumas, M. Solomon, et F. Soumis, 1995. Time constrained routing and scheduling. Dans M. Ball, T. Magnanti, C. Monna, et G. Nemhauser (Rédts.), *Network Routing*, Volume 8 dans *Handbooks in Operations Research and Management Science*, 35–139. Amsterdam, North-Holland : Informs.
- (Desrosiers et al., 1986) J. Desrosiers, Y. Dumas, et F. Soumis, 1986. A dynamic programming solution of the large-scale single-vehicle dial-a-ride problem with time windows. *American Journal of Mathematical Sciences* 6, 301–325.
- (Desrosiers et al., 1984) J. Desrosiers, F. Soumis, et M. Desrochers, 1984. Routing with time windows by column generation. *Networks* 14, 545–565.
- (Dessouky et al., 2003) M. Dessouky, M. Rahimi, et M. Weidner, 2003. Jointly optimizing cost, service, and environmental performance in demand-responsive transit scheduling. *Transportation Research Part D : Transport and Environment* 8, 433–465.
- (Dial, 1995) R. Dial, 1995. Autonomous dial-a-ride transit introductory overview. *Transportation Research C* 3(5), 261–275.
- (Diana et Dessouky, 2004) M. Diana et M. Dessouky, 2004. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research B* 38, 539–557.
- (Dinkelbach, 1967) W. Dinkelbach, 1967. On nonlinear fractional programming. *Management Science* 13(7), 492–498.
- (Drezner, 1996) Z. Drezner, 1996. *Facility location : A survey of applications and methods*. Springer series in Operations Research. Springer-Verlag.

- (Dror, 1994) M. Dror, 1994. Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research* 42(5), 977–978.
- (Dumas et al., 1989) Y. Dumas, J. Desrosiers, et F. Soumis, 1989. Large scale multi-vehicle dial-a-ride systems. Rapport technique, GERAD, École des Hautes Études Commerciales, Montréal.
- (Dumas et al., 1991) Y. Dumas, J. Desrosiers, et F. Soumis, 1991. The pickup and delivery problem with time windows. *European Journal of Operational Research* 54, 7–22.
- (Dumas et al., 1990) Y. Dumas, F. Soumis, et J. Desrosiers, 1990. Optimizing the schedule for a fixed vehicle path with convex inconvenience costs. *Transportation Science* 24, 145–152.
- (Dupuy, 1999) G. Dupuy, 1999. *La dépendance automobile, symptômes, analyses, diagnostic, traitements*. Paris : Anthropos, Collection Villes.
- (Eastman, 1958) W. Eastman, 1958. *Linear programming with pattern constraints*. Thèse de Doctorat, Harvard University.
- (Feillet et al., 2004) D. Feillet, P. Dejax, M. Gendreau, et C. Gueguen, 2004. An exact algorithm for the elementary shortest path problem with resource constraints : application to some vehicle routing problems. *Networks* 44(3), 216–229.
- (Finke, 2002) Finke, G. (Réd.), 2002. *Recherche opérationnelle et réseaux – Méthodes d’analyse spatiale*. Paris : Hermès Sciences Publications.
- (Fu, 2002) L. Fu, 2002. Scheduling dial-a-ride paratransit under time-varying, stochastic congestion. *Transportation Research B* 36, 485–506.
- (Garaix et al., 2007) T. Garaix, D. Feillet, C. Artigues, et D. Josselin, 2007. Vehicle routing problems with alternative paths : an application to on-demand transportation. Rapport technique, UMR Espace et LIA, Université d’Avignon et des Pays de Vaucluse.
- (Garey et al., 1988) M. Garey, R. Tarjan, et G. Wilfong, 1988. One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research* 13, 330–348.
- (Gendreau et al., 1998) M. Gendreau, F. Guertin, J.-Y. Potvin, et R. Séguin, 1998. Dispatching problem with pick-ups and deliveries. Rapport technique CRT-98-10, Centre de Recherche sur les Transport, Université de Montréal.
- (Gørtz, 2006) I. L. Gørtz, 2006. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, Chapitre Hardness of Preemptive Finite Capacity Dial-a-Ride, 200–211. Lecture Notes in Computer Science. Heidelberg : Springer Berlin.
- (Gutiérrez et al., 1998) J. Gutiérrez, A. Monzón, et J. Pinéro, 1998. Accessibility, network efficiency, and transport infrastructure planning. *Environment and Planning A* 30(8), 1337–1350.

- (Gutin et Punnen, 2002) G. Gutin et A. Punnen, 2002. *The Traveling Salesman Problem and its variations*. Dordrecht : Kluwer Academic Publishers.
- (Güting et al., 2000) R. Güting, M. Böhlen, M. Erwig, C. Jensen, N. Lorentzos, M. Schneider, et M. Vazirgiannis, 2000. A foundation for representing and querying moving objects. *Transactions on Database Systems (TODS)* 25(1), 1–42.
- (Harvey et Ginsberg, 1995) W. Harvey et M. Ginsberg, 1995. Limited discrepancy search. Dans les actes de *the fourteenth International Joint Conferences on Artificial Intelligence*, San Francisco, CA, 607–613. Morgan Kaufmann.
- (Hashimoto et al., 2006) H. Hashimoto, T. Ibaraki, S. Imahori, et M. Yagiura, 2006. The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics* 154, 2271–2290.
- (Hauptmeier et al., 2001) D. Hauptmeier, S. Krumke, J. Rambau, et H.-C. Wirth, 2001. Euler is standing in line dial-a-ride problems with precedence-constraints. *Discrete Applied Mathematics* 13, 87–107.
- (Healy et Moll, 1995) P. Healy et R. Moll, 1995. A new extension of local search applied to the dial-a-ride problem. *European Journal of Operational Research* 83, 83–104.
- (Hoogeveen et van de Velde, 1996) J. Hoogeveen et S. van de Velde, 1996. A branch-and-bound algorithm for the single-machine earliness-tardiness scheduling with idle time. *INFORMS Journal of computing* 8, 402–412.
- (Horn, 2002) M. Horn, 2002. Multi-modal and demand-responsive passenger transport systems : a modelling framework with embedded control systems. *Transportation Research A* 36, 167–188.
- (Horn, 2003) M. Horn, 2003. An extended model and procedural framework for planning multi-modal passenger journeys. *Transportation Research B* 37, 641–660.
- (Horn, 2004) M. Horn, 2004. Procedures for planning multi-leg journeys with fixed-route and demand-responsive passenger transport services. *Transportation Research C* 12, 33–55.
- (Houzet et Josselin, 2004) S. Houzet et D. Josselin, 2004. Rapport d'étude de la faisabilité de l'extension du service evolis-gare sur le périmètre de la communauté d'agglomération du grand besançon. Rapport technique, CAGB-ADEME.
- (Hunsaker et Savelsbergh, 2002) B. Hunsaker et M. Savelsbergh, 2002. Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters* 30, 169–173.
- (Ioachim et al., 1993) I. Ioachim, J. Desrosiers, Y. Dumas, et M. Solomon, 1993. A request clustering algorithm for door-to-door handicapped transportation. *Transportation Science* 29, 63–78.
- (Irnich et Desaulniers, 2005) S. Irnich et G. Desaulniers, 2005. Shortest path problems with resource constraints. Dans G. Desaulniers, J. Desrosiers, et M. Solomon (Réds.), *Column Generation*, 33–66. Springer.

- (Isbell et Marlow, 1956) J. Isbell et W. Marlow, 1956. Attrition games. *Naval research logistics quart* 3, 1–99.
- (Jaw et al., 1986) J. Jaw, A. Odoni, H. Psaraftis, et N. Wilson, 1986. A heuristic algorithm for the multi-vehicle many-to-many advance request dial-a-ride problem. *Transportation Research B* 20B, 243–257.
- (Jørgensen et al., 2006) R. Jørgensen, J. Larsen, et K. Bergvinsdottir, 2006. Solving the dial-a-ride problem using genetic algorithms. *Journal of the Operational Research Society*. en ligne.
- (Josselin et Genre-Grandpierre, 2005) D. Josselin et C. Genre-Grandpierre, 2005. Les transports à la demande pour répondre aux nouvelles formes de mobilité. le concept de modulobus. Dans B. Montulet, M. Hubert, C. Jemelin, et S. Schmitz (Réds.), *Mobilités et temporalités*, 151–164. Bruxelles : Facultés Universitaires Saint-Louis.
- (Jung et Haghani, 2000) S. Jung et A. Haghani, 2000. A genetic algorithm for pick-up and delivery problem with time windows. *Transportation Research Record, T.R.B. 1733*, 1–7.
- (Kellerer et al., 2004) H. Kellerer, U. Pferschy, et D. Pisinger, 2004. *Knapsack problems*. Springer.
- (Lacomme et al., 2005) P. Lacomme, C. Prins, et W. Ramdane-Cérif, 2005. Evolutionary algorithms for periodic arc routing problems. *European Journal of Operational Research* 165(2), 535–553.
- (Land et Doig, 1960) A. Land et A. Doig, 1960. An automatic method of solving discrete programming problems. *Econometrica* 28(3), 497–520.
- (Lau et Liang, 2002) H. Lau et Z. Liang, 2002. Pickup and delivery with time windows : Algorithms and test case generation. *International Journal on Artificial Intelligence Tools (Architectures, Languages, Algorithms)* 11(3), 455–472.
- (Li et Lim, 2001) H. Li et A. Lim, 2001. A metaheuristic for the pickup and delivery problem with time windows. Dans les actes de *13th IEEE international conference on tools with artificial intelligence (ICTAI) 2001, Dallas, USA*, 160–170.
- (Lu et Dessouky, 2004) Q. Lu et M. Dessouky, 2004. An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science* 38, 503–514.
- (Lu et Dessouky, 2006) Q. Lu et M. Dessouky, 2006. A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research* 175(2), 672–687.
- (Lübbecke et Desrosiers, 2005) M. Lübbecke et J. Desrosiers, 2005. Selected topics in column generation. *Operations Research* 53(6), 1007–1023.
- (Madsen et al., 1995) O. Madsen, H. Ravn, et J. Rygaard, 1995. A heuristic algorithm for the a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research* 60, 193–208.

- (Mautor et Naudin, 2007) T. Mautor et E. Naudin, 2007. Arcs-states models for the vehicle routing problem with time windows and related problems. *Computers & Operations Research* 34(4), 1061–1084.
- (Melachrinoudis et al., 2007) E. Melachrinoudis, A. Ilhan, et H. Min, 2007. A dial-a-ride problem for client transportation in a health-care organization. *Computers & Operations Research* 34, 742–759.
- (Menger, 1932) K. Menger, 1932. Das botenproblem. *Ergebnisse eines mathematischen kolloquiums* 2, 11–12.
- (Naddef et Rinaldi, 2002) D. Naddef et G. Rinaldi, 2002. Branch-and-cut algorithms for the capacitated VRP. Dans P. Toth et D. Vigo (Réd.), *The Vehicle Routing Problem*, 53–84. Philadelphia : SIAM Monographs on Discrete Mathematics and Application.
- (Nanry et Barnes, 2000) W. Nanry et J. Barnes, 2000. Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B* 34, 107–121.
- (Nikora, 1991) V. Nikora, 1991. Fractal structures of river plan forms. *Water Resources Research* 27(6), 1327–1333.
- (Noyon et al., 2007) V. Noyon, C. Claramunt, et D. Devogele, 2007. A relative representation of trajectories in geographical spaces. *Geoinformatica* 4(11), 479–496.
- (Pankratz, 2003) G. Pankratz, 2003. A grouping genetic algorithm for the pickup and delivery problem with time windows. *OR Spectrum* 27(1), 21–41.
- (Paquette et al., 2007) J. Paquette, J.-F. Cordeau, et G. Laporte, 2007. étude comparative de divers modèles pour le problème de transport à la demande. Rapport technique, CIRRELT, HEC Montréal.
- (Plazanet, 1996) C. Plazanet, 1996. *Enrichissement des bases de données géographiques : analyse de la géométrie des objets linéaires pour la généralisation cartographique (application aux routes)*. Thèse de Doctorat, Université de Marne la Vallée.
- (Prins et al., 2007) C. Prins, C. Prodhon, P. Soriano, A. Ruiz, et R. Wolfer-Calvo, 2007. Solving the capacitated LRP by a cooperative lagrangean relaxation-granular tabu search heuristic. *Transportation Science*. à paraître.
- (Prins et al., 2006) C. Prins, C. Prodhon, et R. Wolfer-Calvo, 2006. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path-relinking. *4OR – A Quarterly Journal of Operations Research* 4(3), 221–238.
- (Psaraftis, 1980) H. Psaraftis, 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science* 14, 130–154.
- (Psaraftis, 1983a) H. Psaraftis, 1983a. Analysis of an $\mathcal{O}(N^2)$ for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science* 17B, 133–145.

- (Psaraftis, 1983b) H. Psaraftis, 1983b. An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science* 17, 351–357.
- (Rekiek et al., 2006) B. Rekiek, A. Delchambre, et H. Saleh, 2006. Handicapped person transportation : An application of the grouping genetic algorithm. *Engineering Application of Artificial Intelligence* 19, 511–520.
- (Righini et Salani, 2005) G. Righini et M. Salani, 2005. New dynamic programming algorithms for the resource constrained elementary shortest path problem. Rapport technique, Dipartimento di Tecnologie dell’Informazione, Università degli Studi di Milano.
- (Righini et Salani, 2006) G. Righini et M. Salani, 2006. Symmetry helps : bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization* 3(3), 255–273.
- (Ropke, 2005) S. Ropke, 2005. *Heuristic and exact algorithms for vehicle routing problems*. Thèse de Doctorat, University of Copenhagen (DIKU).
- (Ropke et al., 2007) S. Ropke, J.-F. Cordeau, et G. Laporte, 2007. Models and branch-and-cut algorithm for pick-up and delivery problems with time windows. *Networks* 49(4), 258–272.
- (Ropke et Pisinger, 2006) S. Ropke et D. Pisinger, 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 40(4), 455–472.
- (Rousseau, 2004) L.-M. Rousseau, 2004. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Chapitre Stabilization Issues for Constraint Programming Based Column Generation, 402–408. Lecture Notes in Computer Science. Heidelberg : Springer Berlin.
- (Rousseau et al., 2004) L.-M. Rousseau, D. Feillet, et M. Gendreau, 2004. New refinements for the solution of vehicle routing problems with column generation. Le Gosier, Guadeloupe (France). TRISTAN 5.
- (Rousseau et al., 2007) L.-M. Rousseau, M. Gendreau, et D. Feillet, 2007. Interior point stabilization for column generation. *Operations Research Letters* 35(5), 660–668.
- (Rousseau et al., 2004) L.-M. Rousseau, M. Gendreau, G. Pesant, et F. Focacci, 2004. Solving VRPTWs with constraint programming based column generation. *Annals of Operations Research* 130(1-4), 199–216.
- (Roy, 1984) S. Roy, 1984. *Routing and Scheduling of Transportation Services for Disabled Persons*. Montréal, Canada : Centre de Recherche sur les Transports, Université de Montréal.
- (Ruland et Rodin, 1997) K. Ruland et E. Rodin, 1997. The pickup and delivery problem : Faces and branch-and-cut algorithm. *Computers and Mathematics with Applications* 33, 1–13.

- (Savelsbergh, 1990) M. Savelsbergh, 1990. En efficient implementation of local search algorithms for constrained routing problems. *European Journal of Operational Research* 47, 75–85.
- (Savelsbergh et Sol, 1995) M. Savelsbergh et M. Sol, 1995. The general pickup and delivery problem. *Transportation Science* 29, 107–121.
- (Savelsbergh et Sol, 1998) M. Savelsbergh et M. Sol, 1998. DRIVE : Dynamic routing of independant vehicles. *Operations Research* 46, 474–490.
- (Schumm, 1963) S. Schumm, 1963. Sinuosity of alluvial channels on the great plains. *Geological Society of America Bulletin* 74, 1089–1100.
- (Semet et Malca, 2006) F. Semet et F. Malca, 2006. Modèles et méthodes pour la gestion de flotte en temps réel. Dans les actes de *Logistique & Transport*, Hammamet, Tunisie. Conférencier Plénier.
- (Sexton et Bodin, 1985a) T. Sexton et L. Bodin, 1985a. Optimizing single vehicle many-to-many operations with desired delivery times : I. Scheduling. *Transportation Science* 19, 378–410.
- (Sexton et Bodin, 1985b) T. Sexton et L. Bodin, 1985b. Optimizing single vehicle many-to-many operations with desired delivery times : II. Routing. *Transportation Science* 19, 411–435.
- (Sexton et Choi, 1986) T. Sexton et Y. Choi, 1986. Pick-up and delivery of partial loads with time windows. *The american journal of mathematical and management sciences* 6, 369–398.
- (Sigurd et al., 2004a) M. Sigurd, D. Pisinger, et M. Sig, 2004a. The pickup and delivery problem with time windows and precedences. Rapport technique, Dpt. of Computer Science, University of Copenhagen.
- (Sigurd et al., 2004b) M. Sigurd, D. Pisinger, et M. Sig, 2004b. Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science* 38, 197–209.
- (Sol, 1994) M. Sol, 1994. *Column generation techniques for pickup and delivery problems*. Thèse de Doctorat, Technische Universitet Eindhoven.
- (Solomon, 1987) M. M. Solomon, 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35(2), 254–265.
- (Stein, 1978) D. Stein, 1978. An asymptotic probabilistic analysis of a routing problem. *Mathematical Operations Research* 3, 89–101.
- (Szwarc et Mukhopadhyay, 1995) W. Szwarc et S. Mukhopadhyay, 1995. Optimal timing schedules in earliness-tardiness single machine sequencing. *Naval Research Logistics* 42, 1109–1114.
- (Teodorovic et Radivojevic, 2000) D. Teodorovic et G. Radivojevic, 2000. A fuzzy logic approach to dynamic dial-a-ride problem. *Fuzzy Sets and Systems* 116, 23–33.

- (Toth et Vigo, 1996) P. Toth et D. Vigo, 1996. Fast local search algorithms for the handicapped persons transportation problem. Dans I. Osman et J. Kelly (Réds.), *Metaheuristics : Theory and applications.*, 677–690. Boston.
- (Toth et Vigo, 1997) P. Toth et D. Vigo, 1997. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science* 31, 60–71.
- (Toth et Vigo, 2002) P. Toth et D. Vigo, 2002. *The vehicle routing problem*. Philadelphia, PA, USA : Society for industrial and Applied Mathematics.
- (Van der Bruggen et al., 1993) L. Van der Bruggen, J. Lenstra, et P. Schuur, 1993. Variable-depth search for the single-vehicle pickup and delivery problem with time windows. *Transportation Science* 27, 298–311.
- (Wiel, 2002) M. Wiel, 2002. *Ville et automobile*. Paris : Descarte Cie.
- (Wilson et Colvin, 1976) N. Wilson et N. Colvin, 1976. Computer control of the rochester dial-a-ride system. Rapport technique, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.
- (Wilson et al., 1971) N. Wilson, J. Sussman, H. Wong, et B. Higonnet, 1971. Scheduling algorithms for dial-a-ride systems. Rapport technique, Urban Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA.
- (Wilson et Weissberg, 1976) N. Wilson et H. Weissberg, 1976. Advanced dial-a-ride algorithms research project : Final report. Rapport technique, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.
- (Wolfler Calvo et Colorni, 2007) R. Wolfler Calvo et A. Colorni, 2007. An effective and fast heuristic for the dial-a-ride problem. *4OR* 5(1), 61–73.
- (Xiang et al., 2006) Z. Xiang, C. Chu, et H. Chen, 2006. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research* 174(2), 1117–1139.
- (Xiang et al., 2007) Z. Xiang, C. Chu, et H. Chen, 2007. The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research* 185(2), 534–551.
- (Xu et al., 2003) H. Xu, Z.-L. Chen, S. Rajagopal, et S. Arunapuram, 2003. Solving a practical pickup and delivery problem. *Transportation Science* 37, 347–364.
- (Zahavi et Ryan, 1980) Y. Zahavi et J. Ryan, 1980. Stability of travel components over time. *Transportation research record* 750, 19–26.
- (Zahavi et Talvitie, 1980) Y. Zahavi et A. Talvitie, 1980. Regularities in travel time and money expenditures. *Transportation research record* 750, 13–19.

Étude et résolution exacte du problème de transport à la demande avec qualité de service.

MOTS-CLEFS : Transport à la demande, problèmes de calcul de tournées de véhicules, génération de colonnes, qualité de service.

RÉSUMÉ : Nous étudions dans cette thèse un problème de construction de tournées de véhicules pour le transport de personnes à la demande (TAD) qui, combinant la souplesse des taxis à la capacité de regroupement des transports en commun, est une voie pour repenser nos pratiques en terme de mobilité.

Après avoir défini puis classé plusieurs critères de qualité de service, nous en sélectionnons trois pour leur représentativité : la minimisation de la distance totale parcourue, la maximisation du taux de remplissage des véhicules et la minimisation du temps perdu en transport. La méthode d'optimisation utilisée est basée sur une approche par décomposition appelée génération de colonnes. Nous nous plaçons dans le cas statique où toutes les demandes sont connues par avance. L'adaptation de cette méthode exacte aux trois critères choisis induit des développements originaux, comme la modélisation du réseau par un p -graphe ou l'optimisation d'une fonction objectif fractionnaire.

Cette étude est intégrée à un projet pluridisciplinaire piloté par des géographes qui a pour sujet d'expérimentation la mise en place d'un TAD opérationnel dans le Pays du Doubs Central (France). Un algorithme de résolution heuristique spécifique a été développé pour cette application. L'intégration des résultats des deux algorithmes à un Système d'Information Géographique permet une analyse des critères de qualité de service et de leurs interactions avec le territoire d'un point de vue géomatique. Il en découle une étude sur la forme des tournées et plus particulièrement sur différentes mesures de leur sinuosité.

Optimization of quality of service for on-demand transportation problems.

KEYWORDS : On-demand transport, vehicle routing problem, column generation, quality of service.

ABSTRACT : In this thesis, we address vehicle routing problems deriving from on-demand transport systems (ODT). Such systems, combining taxi flexibility with public transport system advantages (grouping, price), seem well suited to the new mobility needs.

We define and classify a large set of quality of service criteria. We select three of them among the most representative : the total distance travelled, the vehicle occupancy rate and the passenger wasted-time. We propose branch-and-price solution schemes for the three cases, under the assumption that demands are known in advance. The original quality of service objectives introduce non-standard features into the vehicle routing model, namely a multigraph and a fractional objective function, therefore inducing non-standard algorithms.

This work is part of a multidisciplinary project managed by geographers. The implementation of an operational ODT in the Doubs Central area (France) is used as a testing ground for experimentations. For the practical use of this system, we propose an insertion heuristic in addition to the branch-and-price algorithm. A geomatic analysis of the interactions between quality of service criteria and the geographical area is carried out with a Geographic Information System, with a special focus on route shapes and route sinuosity measures.